Republic of the Philippines
POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
COLLEGE OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CMPE 30052

DATA STRUCTURES AND ALGORITHM

MIDTERMS
LINKED LIST

Submitted by: Signature

Masangcay, Darri Mae A. _____

Roblico, Jay Anton V. _____

Section:

BSCpE 2-2

Submitted to:

Engr. Julius Cansino

Date Submitted:

September 14, 2019

SOURCE CODE:

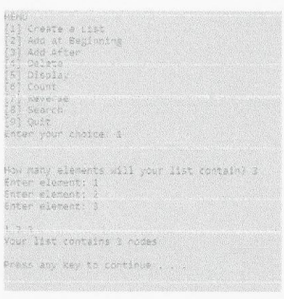
```
#include <iostream>
using namespace std;

struct node
{
      int data;
      node *next;
};

node *head, *newnode, *temp, *hh, *mm, *nodels;
void create(), display(), addbeginning(), addafter(), del(), count(), reverse(),
search();
int data;

void create()
{
      int howmany;
      cout<<"\n\nHow many elements will your list contain? ";
      cin>>howmany;
      while (howmany != 0)
      {
            cout << "ENTER ELEMENT : ";
            cin >> data;
            newnode = new node;
            newnode->data=data;
            newnode->next=NULL;

            if (head == NULL)
            {
                  head = temp = newnode;
            }
            else
            {
                  temp -> next = newnode;
                  temp = newnode;
            }
            howmany--;
      }
      cout<<endl;
      display();
}

void display()
{
      int count;
      temp = head;

      while(temp != NULL)
      {
            cout<< temp -> data <<" ";
            temp = temp -> next;
            count ++;
      }
      cout<<"\nYour list contains "<<count<<" nodes\n\n";
}

void count()
{
      int count;
      temp = head;
      cout<<endl<<endl;
      while(temp != NULL)
      {
            temp = temp -> next;
            count ++;
      }
      cout<<"\nYour list contains "<<count<<" nodes\n";
}

void addbeginning()
{
cout<<"\n\nEnter element to be added ";
cin>>data;
newnode = new node;
```

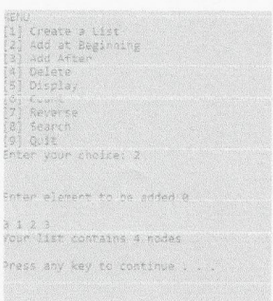
```
newnode -> data = data;
newnode -> next = head;
head = newnode;
cout<<endl;
display();
}

void addafter()
{
      int position; int i=1; int count;
      cout<<"\n\nEnter after position: ";
      cin>>position;

      temp = head;
      while(temp != NULL)
      {
            temp = temp -> next;
            count ++;
      }

if(position>count)
{
      cout<<"Invalid position";
}
else
{
      temp = head;
      while(i<position)
      {
            temp = temp->next;
            i++;
      }
      newnode = new node;
}

cout<<"Enter element to be added: ";
cin>>data;
newnode -> data = data;
newnode -> next = temp -> next;
temp -> next = newnode;
cout<<endl;
display();
}

void del()
{
      int delposition = 0;
      bool checker = false;

      if (head == 0)
      {
            cout<<"SORRY, YOU CAN NOT DELETE FROM AN EMPTY LIST.";
      }
            cout<<"\n\nWhat is the element you want to remove from the list? ";
            cin>>data;
            nodels = head;

            while (nodels != 0)
            {
                  delposition++;
                  if (nodels->data == data)
                  {
                        checker = true; break;
                  }
                  nodels = nodels -> next;
            }

            if(!checker)
            {
            cout<<"Sorry, your input is not in the list."<<endl;
            }
      node *hh = new node;
      node *mm = new node;
      hh = head;
```

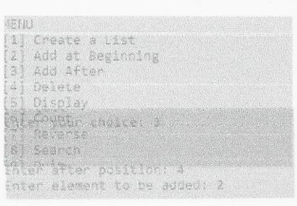
```
	for(int i=1;i!=delposition; i++)
	{
		mm=hh;
		hh=hh->next;
	}
	mm->next=hh->next;
	cout<<endl;
	display();
}

void reverse()
{
	hh = head;
	mm = NULL;
	newnode = NULL;
	while(hh!=NULL)
	{
		newnode = hh->next;
		hh->next=mm;
		mm = hh;
		hh = newnode;
	}
	head=mm;
	cout<<endl;
	display();
}

void search()
{
	//Node *temp;
	int searchposition=0;
	bool checker=false;
	if(checker!=0)
	{
		cout<<"Sorry, the list contains no elements."<<endl;
	}
	cout<<"\n\nWhat element do you want to find in the list? ";
	cin>>data;
	nodels=head;
	while(nodels!=0)
	{
		searchposition++;
		if(nodels->data==data)
		{
			checker=true;
			cout<<"Element "<<data<<" is found at position
"<<searchposition<<endl;
		}
		nodels=nodels->next;
	}
	if(!checker)
	{
		cout<<"Element "<<data<<" is not in the list."<<endl;
	}
}

void Menu()
{
	cout<<"MENU"<<endl;
	cout<<"[1] Create a List"<<endl;
	cout<<"[2] Add at Beginning"<<endl;
	cout<<"[3] Add After"<<endl;
	cout<<"[4] Delete"<<endl;
	cout<<"[5] Display"<<endl;
	cout<<"[6] Count"<<endl;
	cout<<"[7] Reverse"<<endl;
	cout<<"[8] Search"<<endl;
	cout<<"[9] Quit"<<endl;
}

int main()
{
	int choice = 0;
	do
	{
```

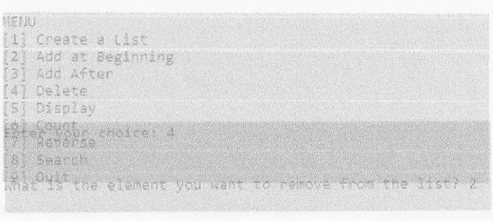
```
                system("cls");
                Menu();
                cout<<"Enter your choice: ";
                cin>>choice;

                switch (choice)
                {
                      case 1:
                            create(); system("pause"); break;
                      case 2:
                            addbeginning(); system("pause"); break;
                      case 3:
                            addafter(); system("pause"); break;
                      case 4:
                            del(); system("pause"); break;
                      case 5:
                            display(); system("pause"); break;
                      case 6:
                            count(); system("pause"); break;
                      case 7:
                            reverse(); system("pause"); break;
                      case 8:
                            search(); system("pause"); break;
                      case 9:
                            cout<<"\n\nSystem will now be closed. Thank you!!!!!
"<<endl<<endl; system("pause"); break;
                     default:
                            cout<<"Invalid choice"<<endl;
               }
      } while (choice != 9);
      return 0;
}
```

SAMPLE OUTPUT:









SOURCE CODE:

```
#include<iostream>
#include <stdlib.h>
#include <cstdlib>
using namespace std;

struct Node
{
   int number;
   Node *next;
};
Node *HEAD = NULL;
void display(Node* head);
void insertatfirst();
void insertatnthnode(int loc, int value);
void search(Node *head);
Node* createLinkedList(int n);
void delet(int n);
void menu();
void displaynumofelements(Node* Head);
void reverselist (Node* Head);
int main (){

        menu();
}
void menu(){
        cout << "1. Create" << "\n"
                <<      "2. Add at Beginning"
<< "\n"
                << "3. Add after" << "\n"
                << "4. Delete" << "\n"
                << "5. Display" << "\n"
                << "6. Display" << "\n"
                << "7. Reverse" << "\n"
                << "8. Search" << "\n"
                << "9. Quit" << "\n";
        int choice, nodenum, nvaluedelete,
nvalueinsert, ninsertloc,valtosearch;
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice){
                case 1:
                {
                        cout << "How many
nodes?: " << endl;
                        cin >> nodenum;
                        HEAD =
createLinkedList(nodenum);
                        display(HEAD);
                        cout << endl;
                        break;
                }
                case 2:
                {
                        insertatfirst();
                        break;
                }
                case 3:{
                        cout << "Enter value to
insert: ";
                        cin >> nvalueinsert;
                        cout << "Enter the
location of where you want to insert the value
after: ";
                        cin >> ninsertloc;

        insertatnthnode(ninsertloc,
nvalueinsert);
                        break;
                }
                case 4:{
                        cout << "Enter what
value to delete: ";
                        cin >> nvaluedelete;
                        delet(nvaluedelete);
                        display(HEAD);
                        break;
                }
                case 5:{
                        display(HEAD);
                        break;
                }
                case 6:{

        displaynumofelements(HEAD);
                        break;
                }
                case 7:{
                        reverselist(HEAD);
                        break;
                }
                case 8:{
                        search(HEAD);
                        break;
```

```
                }
                case 9:{
                        exit(0);
                }}
        menu();
}
Node* createLinkedList(int n){

        Node *Head = new Node;
        Head = NULL;
        Node *temp;
        temp = NULL;
        temp = new Node;
        Node *it = new Node;
        it = NULL;
        for(int i=0;i < n;i++){
                temp =
(Node*)malloc(sizeof(Node));
                cout << "Enter the number for
node number " << i+1 << " " << ": ";
                cin >> temp->number;
                temp->next=NULL;
                if (Head == NULL){
                        Head = temp;
                }
                else{
                        it=Head;
                        while(it->next != NULL)
                        it = it->next;
                        it->next = temp;
                }
        }
        return Head;
        menu();

}
void display(Node* head){
        Node*it = head;
        while(it != NULL){
                cout << it->number << "-> ";
                it = it->next;
        }
                cout << "NULL" << endl;
}
void insertatfirst(){
        Node* ins = new Node;
        if(HEAD == NULL){
                        int nodenum;
                        cout << "List is empty,
please create a node first." << endl;
                        Node*HEAD = NULL;
                        cout << "How many
nodes?: " << endl;
                        cin >> nodenum;
                        HEAD =
createLinkedList(nodenum);
                        display(HEAD);
                        cout << endl;
                        }
        else{
                int insertData;
                cout << "Enter the integer
value: ";
                cin >> insertData;
                Node *p = new Node;
                p->number = insertData;
                p->next = HEAD;
                HEAD = p;
        }
        display(HEAD);
}
void delet(int n){
        if(HEAD->number == n){
                Node *current = new Node;
                current = HEAD;
                HEAD = HEAD->next;
                delete current;
        }
        else{
                                Node *current
= new Node;
                                Node *prev =
/*this->*/HEAD;
                current = HEAD->next;
                while(current != NULL) {
        if(current->number == n)
                                {
                break;
        }
        else
                                {
                prev = current;
                current = current->next;
        }
        }
        if(current == NULL)
```

```
                {
        cout << "The element is not found!\n";
        }
                else
                {
        prev->next = current->next;
        delete current;
        }}
}

void insertatnthnode(int loc, int value){
        Node *p1 = new Node;
        Node *p3 = new Node;
        Node *p = new Node;
        p3 = HEAD;
        for(int i = 1; i < loc + 1; i++)
        {
                p1 = p3;
                p3 = p3->next;
        }
        p->number = value;
        p1->next = p;
        p->next = p3;
        display(HEAD);
}
void displaynumofelements(Node* Head){
        Node*it = Head;
        int i = 0;
        while(it != NULL){
                it = it->next;
                i += 1;
        }
        cout << i << endl;
                current = HEAD->next;
                while(current != NULL) {
        if(current->number == n)
                        {
                                cout <<
"The number is in the list\n";
                break;
        }
        else
                        {
                prev = current;
                current = current->next;
        }}
        if(current == NULL)
                {
        cout << "The element is not found!\n";
        } }
}
void reverselist(Node *Head){
        Node *ptr1 =new Node;
        Node *ptr2 = new Node;
        Node *ptr3 = new Node;
  if (Head == NULL)
  {
    cout<<"List is empty"<<endl;
    return;
  }
  if (Head->next == NULL)
  {
    return;
  }
  ptr1 = Head;
  ptr2 = ptr1->next;
```

Polytechnic University of the Philippines
College of Engineering
Computer Engineering Department

CMPE 30052

Data Structures and Algorithms

Midterm Project

Linked-list

BSCPE 2-2

Submitted by:

NAME	Signature
DECANO, Rainier M.	
INTERIOR, Nicolle S.	

Submitted to:

Engr. JULIUS CANSINO

September 14, 2019

SOURCE CODE:

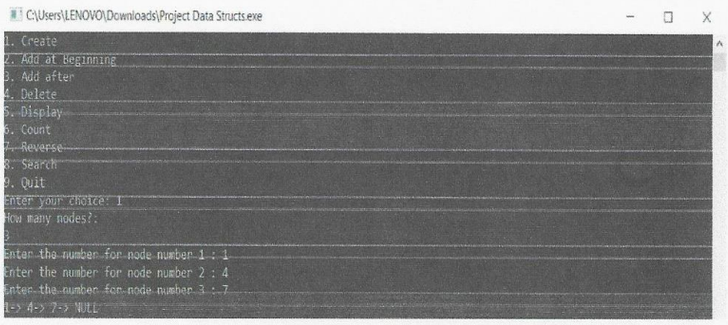
```
#include<iostream>
#include <stdlib.h>
#include <cstdlib>
using namespace std;

struct Node
{
   int number;
   Node *next;
};
Node *HEAD = NULL;
void display(Node* head);
void insertatfirst();
void insertatnthnode(int loc, int value);
void search(Node *head);
Node* createLinkedList(int n);
void delet(int n);
void menu();
void displaynumofelements(Node* Head);
void reverselist (Node* Head);
int main (){

        menu();
}
void menu(){
        cout << "1. Create" << "\n"
                <<      "2. Add at Beginning"
<< "\n"
                << "3. Add after" << "\n"
                << "4. Delete" << "\n"
                << "5. Display" << "\n"
                << "6. Display" << "\n"
                << "7. Reverse" << "\n"
                << "8. Search" << "\n"
                << "9. Quit" << "\n";
        int choice, nodenum, nvaluedelete,
nvalueinsert, ninsertloc,valtosearch;
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice){
                case 1:
                {
                        cout << "How many
nodes?: " << endl;
                        cin >> nodenum;
                        HEAD =
createLinkedList(nodenum);
                        display(HEAD);
                        cout << endl;
                        break;
                }
                case 2:
                {
                        insertatfirst();
                        break;
                }
                case 3:{
                        cout << "Enter value to
insert: ";
                        cin >> nvalueinsert;
                        cout << "Enter the
location of where you want to insert the value
after: ";
                        cin >> ninsertloc;

                insertatnthnode(ninsertloc,
nvalueinsert);
                        break;
                }
                case 4:{
                        cout << "Enter what
value to delete: ";
                        cin >> nvaluedelete;
                        delet(nvaluedelete);
                        display(HEAD);
                        break;
                }
                case 5:{
                        display(HEAD);
                        break;
                }
                case 6:{

                displaynumofelements(HEAD);
                        break;
                }
                case 7:{
                        reverselist(HEAD);
                        break;
                }
                case 8:{
                        search(HEAD);
                        break;
```

```
                }
                case 9:{
                        exit(0);
                }}
        menu();
}
Node* createLinkedList(int n){

        Node *Head = new Node;
        Head = NULL;
        Node *temp;
        temp = NULL;
        temp = new Node;
        Node *it = new Node;
        it = NULL;
        for(int i=0;i < n;i++){
                temp =
(Node*)malloc(sizeof(Node));
                cout << "Enter the number for
node number " << i+1 << " " << ": ";
                cin >> temp->number;
                temp->next=NULL;
                if (Head == NULL){
                        Head = temp;
                }
                else{
                        it=Head;
                        while(it->next != NULL)
                        it = it->next;
                        it->next = temp;
                }
        }
        return Head;
        menu();

}
void display(Node* head){
        Node*it = head;
        while(it != NULL){
                cout << it->number << "-> ";
                it = it->next;
        }
                cout << "NULL" << endl;
}
void insertatfirst(){
        Node* ins = new Node;
        if(HEAD == NULL){
                        int nodenum;
                        cout << "List is empty,
please create a node first." << endl;
                        Node*HEAD = NULL;
                        cout << "How many
nodes?: " << endl;
                        cin >> nodenum;
                        HEAD =
createLinkedList(nodenum);
                        display(HEAD);
                        cout << endl;
                        }
        else{
                int insertData;
                cout << "Enter the integer
value: ";
                cin >> insertData;
                Node *p = new Node;
                p->number = insertData;
                p->next = HEAD;
                HEAD = p;
        }
        display(HEAD);
}
void delet(int n){
        if(HEAD->number == n){
                Node *current = new Node;
                current = HEAD;
                HEAD = HEAD->next;
                delete current;
        }
        else{
                        Node *current
= new Node;
                        Node *prev =
/*this->*/HEAD;
                current = HEAD->next;
                while(current != NULL) {
        if(current->number == n)
                        {
                break;
        }
        else
                        {
                prev = current;
                current = current->next;
        }
        }
        if(current == NULL)
```

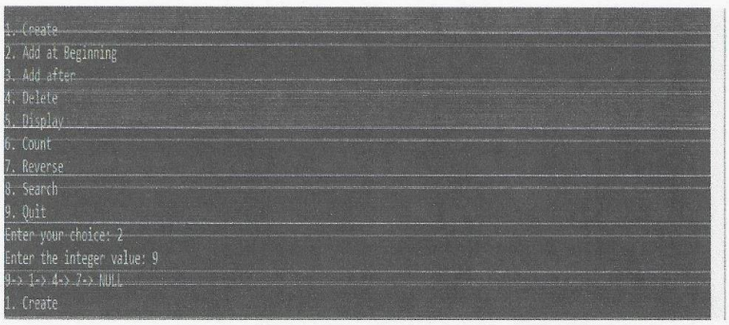
```
                        {
        cout << "The element is not found!\n";
        }
                        else
                        {
        prev->next = current->next;
        delete current;
        }}
}

void insertatnthnode(int loc, int value){
                Node *p1 = new Node;
                Node *p3 = new Node;
                Node *p = new Node;
                p3 = HEAD;
                for(int i = 1; i < loc + 1; i++)
                {
                        p1 = p3;
                        p3 = p3->next;
                }
                p->number = value;
                p1->next = p;
                p->next = p3;
                display(HEAD);
}
void displaynumofelements(Node* Head){
        Node*it = Head;
        int i = 0;
        while(it != NULL){
                it = it->next;
                i += 1;
        }
        cout << i << endl;
}
void search(Node *head){

        int n;
        cout << "Enter value to search: " <<
endl;
        cin >> n;
if(HEAD->number == n){
        cout << "The number is in the list";
        }
        else{
                                Node *current
= new Node;
                                Node *prev =
/*this->*/HEAD;
                current = HEAD->next;
                while(current != NULL) {
        if(current->number == n)
                                {
                                        cout <<
"The number is in the list\n";
                break;
        }
        else
                                {
                prev = current;
                current = current->next;
        }}
        if(current == NULL)
                        {
        cout << "The element is not found!\n";
        } }
}
void reverselist(Node *Head){
        Node *ptr1 =new Node;
        Node *ptr2 = new Node;
        Node *ptr3 = new Node;
  if (Head == NULL)
  {
     cout<<"List is empty"<<endl;
     return;
  }
  if (Head->next == NULL)
  {
     return;
  }
  ptr1 = Head;
  ptr2 = ptr1->next;
  ptr3 = ptr2->next;
  ptr1->next = NULL;
  ptr2->next = ptr1;
  while (ptr3 != NULL)
  {
     ptr1 = ptr2;
     ptr2 = ptr3;
     ptr3 = ptr3->next;
     ptr2->next = ptr1;
  }
  HEAD = ptr2;
  display(HEAD);
}
```

OUTPUT:

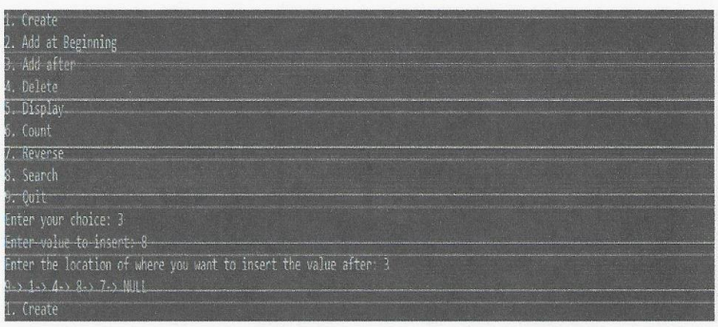
CREATE:



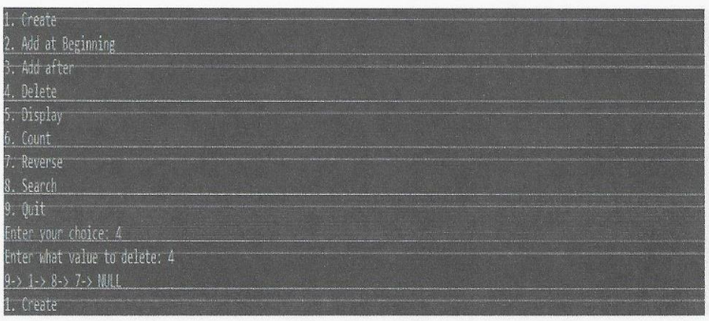
ADD AT BEGINNING:



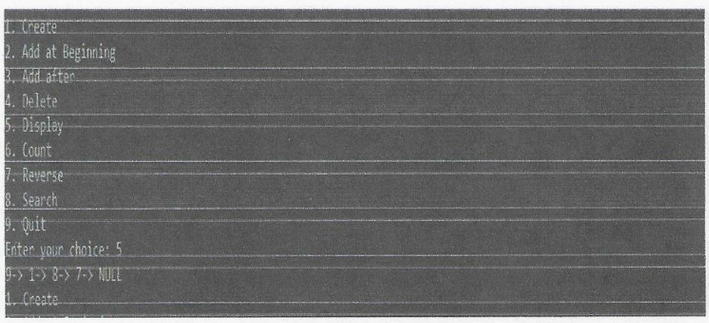
ADD AFTER:



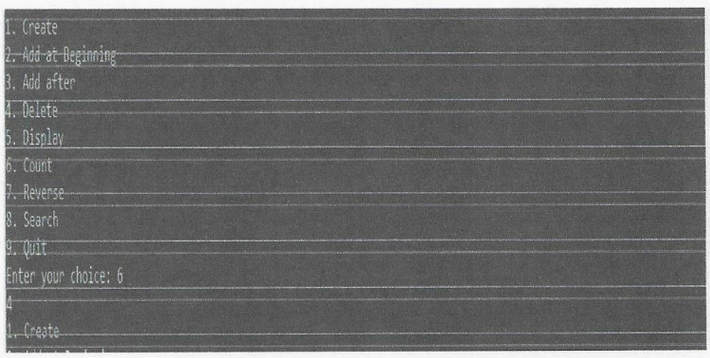
DELETE:



DISPLAY:



COUNT:



REVERSE:

```
1. Create
2. Add at Beginning
3. Add after
4. Delete
5. Display
6. Count
7. Reverse
8. Search
9. Quit
Enter your choice: 7
7-> 8-> 1-> 9-> NULL
1. Create
```

SEARCH:

```
Enter your choice: 8
Enter value to search:
9
The number is in the list
1. Create
2. Add at Beginning
3. Add after
4. Delete
5. Display
6. Count
7. Reverse
8. Search
9. Quit
Enter your choice: 8
Enter value to search:
2
The element is not found!
1. Create
```

POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
COLLEGE OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Computer ENGINEERING TECHNOLOGY 1 PROJECT

Engineer Orland Tubola

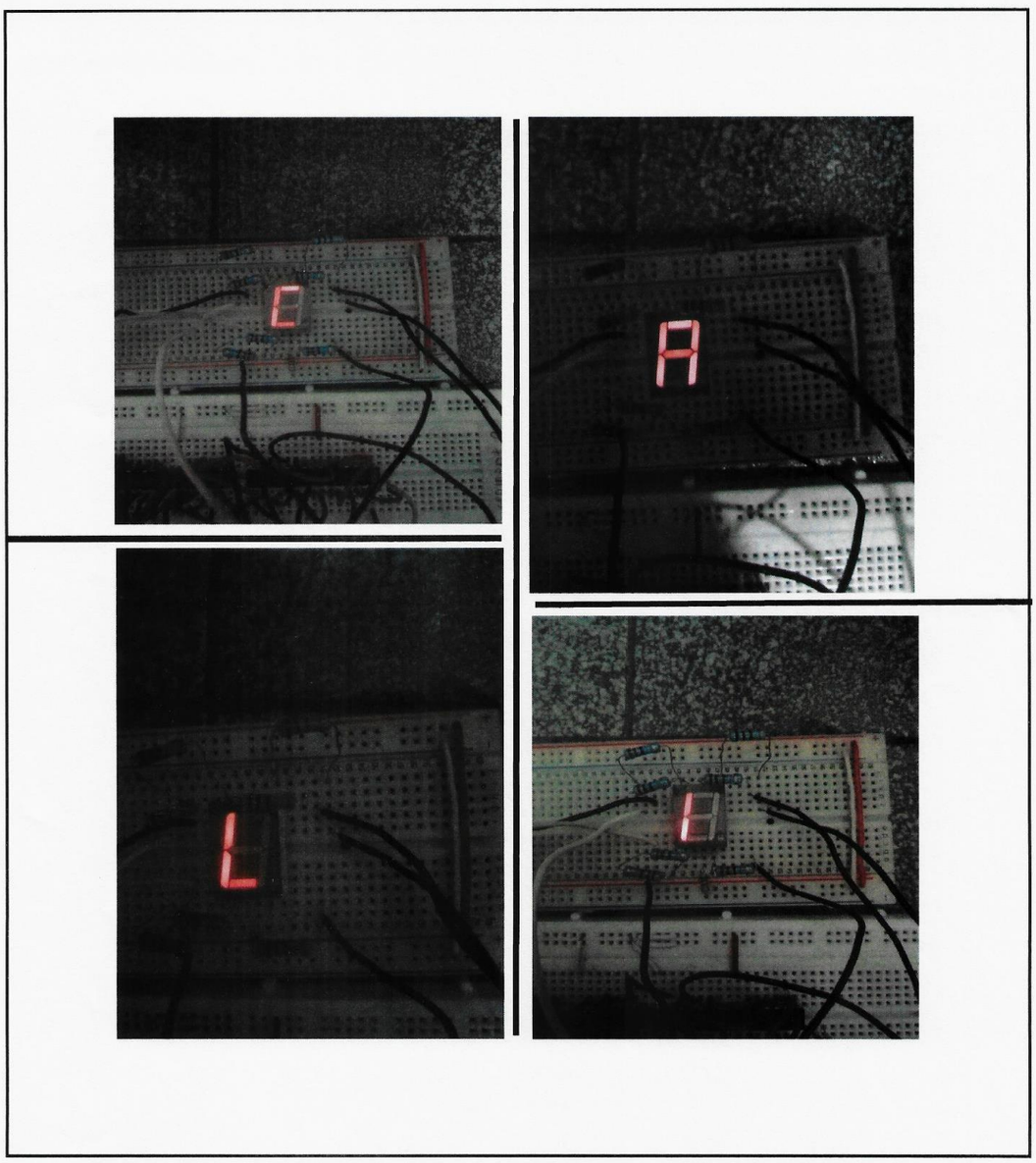
Submitted by:

Agoncillo, Guia Angela
Almonte, Hillary
Andrada, Ali
Cuevas, Janella
Del Rosario, Maubrey
Galarosa, Angelu
Ladion, Vince
Mangabay, Kyle
Nombrado, Paul Joseph
Ricafort, Noel
Sison, Jaeson
Velarde, Prince Charles
Alejo, Alvin Dale
Anciado, Jericho
Bocanog, Verlanne
Decano, Rainier
Gajelan, Mafe
Janeo, Janelle
Magadia, Sammael
Matibag, Kyla Mae
Puno, Erika
Rivera, Stephanie Eurice
Teoxon, Emmanuel
Villas, Bruce Jared

TRUTH TABLE

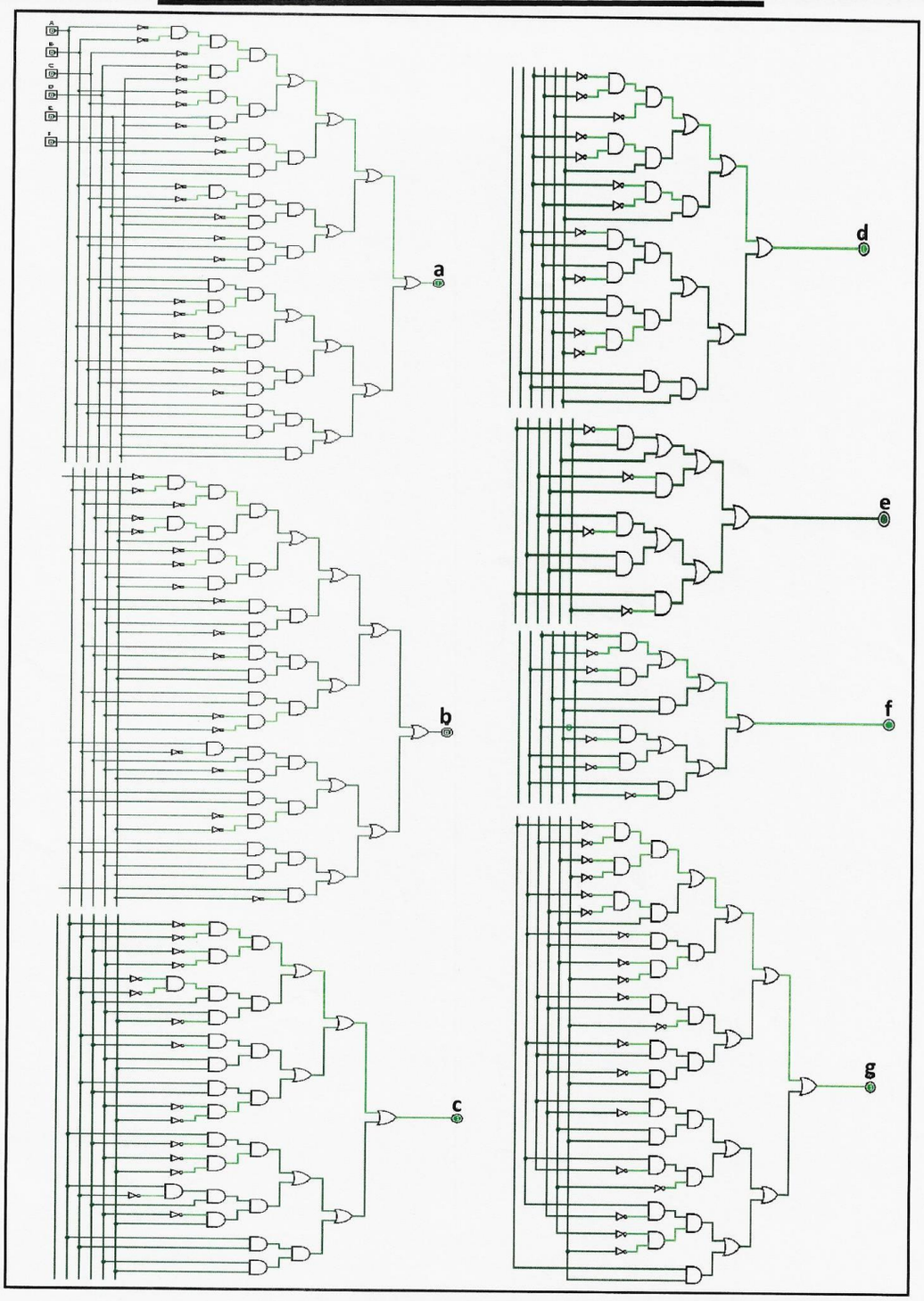
A	B	C	D	E	F	a	b	c	d	e	f	g
0	0	0	0	0	0	1	0	1	1	0	1	1
0	0	0	0	0	1	0	1	1	1	1	1	0
0	0	0	0	1	0	1	1	0	0	1	1	1
0	0	0	0	1	1	1	0	0	1	1	1	1
0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	1	1	0	0	1	1	1	0
0	0	0	1	1	0	1	1	1	0	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	0
0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	1	1	0	0	0	1	1	1
0	0	1	0	1	0	0	0	0	0	1	0	1
0	0	1	0	1	1	1	1	1	0	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1
0	0	1	1	0	1	0	0	0	0	1	1	0
0	0	1	1	1	0	0	0	0	1	1	1	0
0	0	1	1	1	1	0	0	0	0	1	1	0
0	1	0	0	0	0	1	0	1	1	0	1	1
0	1	0	0	0	1	0	0	0	1	1	1	1
0	1	0	0	1	0	0	0	0	0	1	1	0
0	1	0	0	1	1	0	1	0	1	1	1	0
0	1	0	1	0	0	1	0	0	1	1	1	1
0	1	0	1	0	1	0	1	1	0	1	1	1
0	1	0	1	1	0	1	1	0	0	1	1	1
0	1	0	1	1	1	0	0	0	0	1	1	0
0	1	1	0	0	0	1	1	1	0	1	1	1
0	1	1	0	0	1	0	0	0	1	1	1	0
0	1	1	0	1	0	0	0	0	0	1	1	0
0	1	1	0	1	1	0	1	1	1	1	0	1
0	1	1	1	0	0	1	1	1	1	1	1	0
0	1	1	1	0	1	1	0	0	1	1	1	0
0	1	1	1	1	0	0	0	0	0	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1	1	1	1	1	0
1	0	0	0	0	1	1	0	1	1	0	1	1

ACTUAL PICTURE



(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

SCHEMATIC DIAGRAM



(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

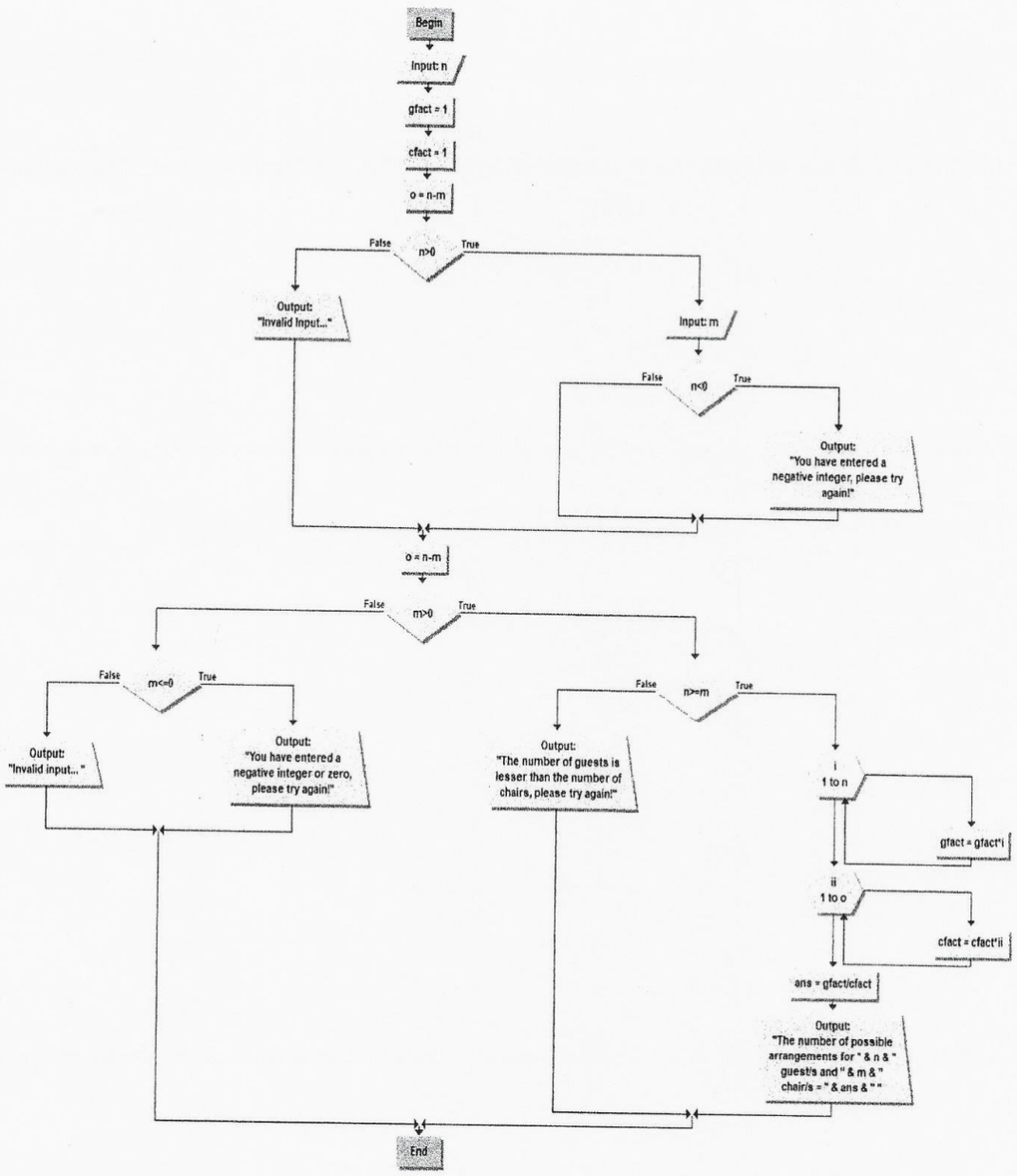
Programming, Logic, and Design

(Written Report)
Ma. Leona S. Khan

Scarlet Raven
"The ravens are Prog-roaming in the night sky"
BSCpE 1-1
David Jan A. Afalla
Janella T. Cuevas
Vince Jeremy T. Ladion
Jay Anton V. Roblico
Mark Christer Salamante

3Θ. A program that calculates the number of possible arrangements for any number of guests and any number of chairs. (Assume there will never be fewer guests than chairs).

Flow chart made in Visual Logic application:



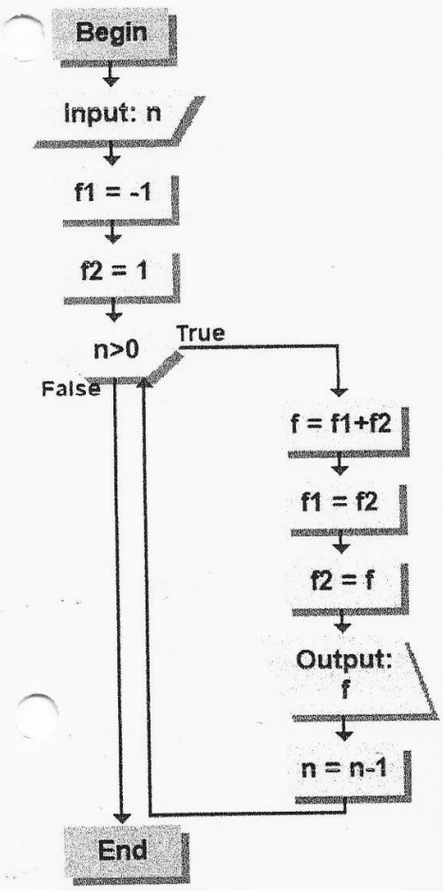
Source code made with DevC++ application:

```
#include <iostream>
using namespace std;

int main()
{
    int i, ii, n, m, o, ans, gfact = 1, cfact = 1;

    cout <<"Enter the number of guests: ";
    cin >> n;
    if (n>0)
    {
        cout <<"Enter the number of chairs (must be lesser than the number of guest): ";
        cin >> m;
    }
    else if (n<0)
    {
        cout << "You have entered a negative integer, please try again!";
        return 0;
    }
    else
    {
        cout << "Invalid Input. . .";
        return 0;
    }
    o = n-m;

    if (m>0)
    {
        if (n>m)
        {
            for (i = 1; i<=n; ++i)
            {
                gfact *= i;
            }
            for (ii= 1; ii<=o; ++ii)
            {
                cfact *= ii;
            }
                ans = gfact / cfact;
                cout << "The number of possible arrangements for " <<n<< " guest/s and " <<m<< " chair/s = " <<ans<< "  " <<endl;
        }
        else if (n<m)
        {
            cout << "The number of guest is lesser than the number of chairs, please try again!";
        }
        else if (n=m)
        {
            for (i = 1; i<=n; ++i)
            {
                gfact *= i;
            }
            cout << "The number of possible arrangements for " <<n<< " guest/s and " <<m<< " chair/s = "<<gfact<<endl;
        }
    }
    else if (m<=0)
    {
        cout << "You have entered a negative integer or zero, please try again!";
    }
    else
    {
        cout << "Invalid Input. . .";
    }
        return 0;
}
```

```
C:\Users\Jay Anton V. Roblico\Desktop\Programming\Long Quiz\SourceCodes\G...
Enter the number of guests: 6
Enter the number of chairs (must be lesser than the number of guest): 4
The number of possible arrangements for 6 guest/s and 4 chair/s = 360

--------------------------------
Process exited after 5.149 seconds with return value 0
Press any key to continue . . .
```

7e. A program that will simulate the Fibonacci Series from the start to the nth term.

Flow chart made with Visual Logic application:

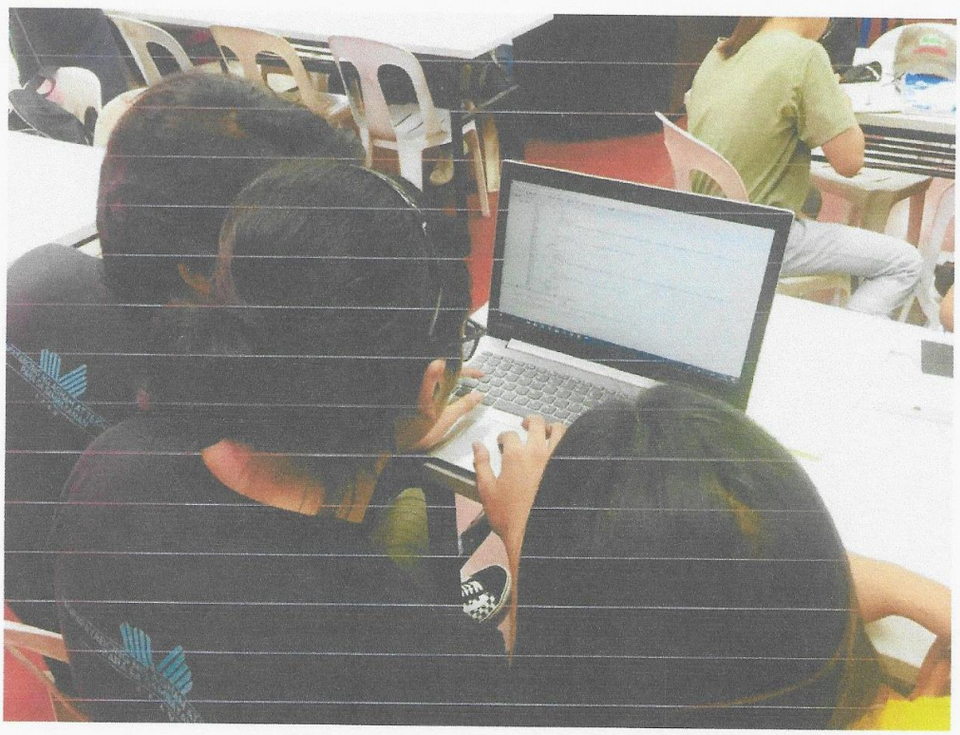


Source code made with DevC++ application:

```
1   #include<iostream>
2
3   using namespace std;
4
5   int main()
6   {
7       int n, f, f1=-1, f2=1;
8
9       cout << "Enter The Number Of Terms: ";
10      cin >> n;
11      cout<<"The Fibonacci Series is:";
12
13      while(n>0)
14      {
15          f=f1+f2;
16          f1=f2;
17          f2=f;
18          cout<<"  \n"<<f;
19          n--;
20      }
21
22      return 0;
23  }
```

```
C:\Users\Jay Anton V. Roblico\Desktop\Programming\Lo...
Enter The Number Of Terms: 15
The Fibonacci Series is:
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377

Process exited after 1.721 seconds with return value 0
Press any key to continue . . .
```


(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA



A group of students performing their activity in computer programming subject

A student reporting a topic in front of his class

62

COEN 3193 - Data Communications
2nd Semester / AY 2018-2019

LONG QUIZ NO. 1

Introduction to Data Communications and Data Communication Codes

Name	
Student Number	
Section/Day(s)/Time	
Faculty Name	Engr. John Dela Cruz
Date of Exam	January 08, 2019

GENERAL INSTRUCTION

1. This is a **2-HOUR EXAM.** Please use your time wisely.
2. Everyone is obliged to use **BLACK OR BLUE INK BALLPENS ONLY. PENCILS, FRICTION PENS, FOUNTAIN PENS, AND OTHER TYPES OF PEN WITH HEAVY BLOTTING ARE NOT ALLOWED.**
3. Read and understand the questions very carefully. Questions should be raised directly to the instructor/proctor. **ASKING QUESTIONS TO OTHER EXAMINEES IS CONSIDERED AS CHEATING.**
4. Please make sure your handwriting is a readable one, especially for computations. **UNRECOGNIZABLE HANDWRITING MAY LEAD TO A WRONG ANSWER.**
5. **INCORRECT VARIABLE OR UNIT CORRESPONDS TO ZERO POINT.**
6. Honesty is the best policy. **CHEATING WILL CORRESPOND TO A ZERO SCORE AND FAILING GRADE, AND WILL BE SUBJECTED TO DISCIPLINARY ACTION AS PRESCRIBED IN THE STUDENT HANDBOOK.**
7. **INDEX CARDS ARE NOT ALLOWED.**
8. **USE THE BACK PART OF THE QUESTIONNAIRE AS YOUR SCRATCH PAPER.**

LaTeX by JRDC

EN 3193 - Data Communications
Long Quiz No. 1

PUP College of Engineering
AY 2018-2019 2nd Semester

Part I. Identification. Identify the word or group of words that is being described. Write your answer at the left side of the number. ***Abbreviations are NOT ALLOWED. Surnames only are also NOT ALLOWED.***. *(2 pts. each, RIGHT minus 0.5 × WRONG)*

Telegraph 1. This device was invented in 1837, and was considered as the forefather of electronic data transmission.

Samuel F.B. Morse 2. Who invented the device above?

transducer 3. This is a general term of a device that converts a physical quantity into electrical signal.

Channel 4. This part of the communication system is greatly affected by noise.

R-11 W-4 Protocol 5. This is a set of rules that govern data communications.

Standard 6. These are agreed rules or format that has been approved by a recognized organization.

Timing 7. This refers to the variation on the arrival time of each data packets.

~~Entrophy~~ Entropy 8. It is the probabilistic behavior of a source of information, which is also a measure of the average information content per symbol.

Information technology 9. This is the mathematical treatment of the concepts, parameters, and rules governing the transmission of messages.

Source coding theorem 10. This theorem establishes the limit to possible data compression, and the operation meaning of entropy.

David Huffman 11. He published the paper "A Method of Construction of Minimum-Redundancy Codes" in 1952.

Channel capacity 12. This is defined as the intrinsic ability of a channel to convey information.

Baudot code 13. The first character coding scheme used in telecommunications.

American standard code for information interchange 14. The first fixed-length coding scheme used in telecommunications.

~~FBDC~~ EBCDIC 15. This fixed-length coding scheme was mainly used on IBM mainframe and IBM midrange computer OS. *(NOTE: Acronym is allowed for this number.)*

Part II. Enumeration. Enumerate the possible correct answers on each item. The maximum number of answers and the number of point per item are indicated. ***Exceeding to the maximum number will forfeit all the answers, whether they are correct or wrong.***

1. Five (5) elements of the basic communication system block diagram. *(10 pts.)*

8

1) Source
2) Transmitter
3) Channel
4) ~~Receiver~~ Noise
5) Reciever

2. Four (4) fundamental characteristics of data communications. *(8 pts.)*

8

1.) Delivery
2.) accuracy
3.) timeliness
4) Jitter

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

N 3193 - Data Communications
Long Quiz No. 1

PUP College of Engineering
AY 2018-2019 2nd Semester

3. Three (3) main types of data flow. *(6 pts.)*
   1.) Syntax
   2.) Semantic
   3.) timing

4. Two (2) types of standards. *(4 pts.)*
   1.) De jure standards
   2.) De facto standard

4

5. Three (3) recognized standards organizations for data communications. ***NO Abbreviations allowed*** *(9 pts.)*
   1.) Internet Engineering Task force
   2.) ~~International Electron~~ International Organization for Standardization
   3.) International electronical comission

6

6. Two (2) elements of a Morse code. *(4 pts.)*
   1.) dot
   2.) dashes

4

Part III. Tables. Determine the equivalent ASCII and EBCDIC Binary strings for each character indicated. ***Your answers shall be grouped per nibble starting from the LSB. For ASCII, zero (0) shall be used as the MSB.*** *(3 pts. per correct answer)*

Character	ASCII Binary Equivalent	EBCDIC Binary Equivalent
J		0110 0000
q		
S		
4		
7		
Y		
g		
m		
P		
L		

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

N 3193 - Data Communications
Long Quiz No. 1

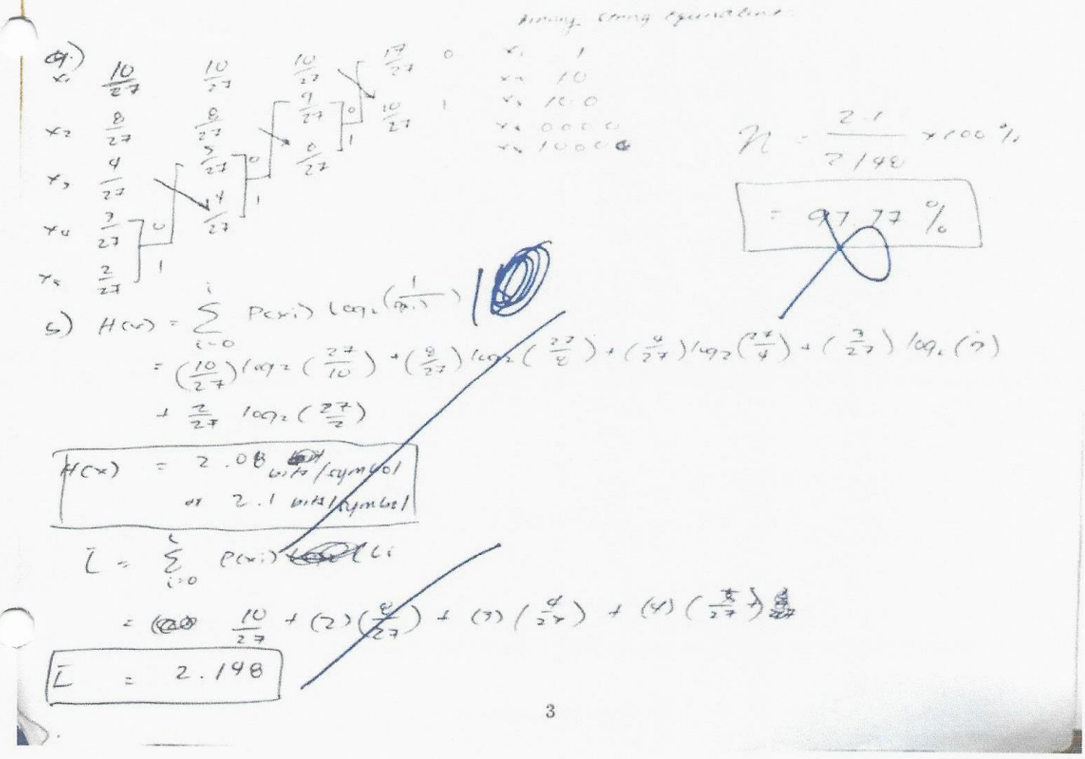
PUP College of Engineering
AY 2018-2019 2nd Semester

Part IV. Problem Solving. Answer the following problems. All solutions should be shown in a logical way. ***Always box the final and required answers. NO partial points for solution.***

Problem 1. A new character encoding scheme will be used in deciphering data sent through a highly-classified data communication system set-up by FBI. They will use variable-length coding and will be based on Huffman's technique. Unfortunately, only five (5) characters will be used.

a. Complete the table shown below. Show your solution on how did you come up with the codes. *(15 pts.)*
b. Compute for the entropy, average code-word length, and the efficiency of the coding scheme. *(15 pts.)*

Char. no.	Char.	Probability	Binary String Equiv.
x_1	✱	$\frac{4}{27}$	1
x_2	○	$\frac{8}{27}$	10
x_3	§	$\frac{2}{27}$	100
x_4	😊	$\frac{10}{27}$	0000
x_5	o|o	$\frac{3}{27}$	1000



(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

EN 3193 - Data Communications
Long Quiz No. 1

PUP College of Engineering
AY 2018-2019 2nd Semester

Problem 2. Using Shannon-Fano Coding, determine the equivalent binary string of the word HABOPEN using the table of probability below. HINT: Arrange from highest probability to lowest probability. Also, what is the efficiency of the coding scheme? *(25 pts., Group the binary strings per nibble starting from the LSB.)*

Character	Probability
A	0.1086
B	0.1329
E	0.1525
H	0.0526
N	0.2419
O	0.2013
P	0.1102

$P(x_i)$

N	0.2419	0	00
O	0.2013	1	01
E	0.1525	0	100
B	0.1329	1	101
P	0.1102	0	110
A	0.1086	0	1110
H	0.0526	1	1111

$$H(x) = \sum_{i=0}^{i} P(x_i) \log_2 \left(\frac{1}{P(x_i)}\right)$$

$$= (0.2419) \log_2 \left(\frac{1}{0.2419}\right) + (0.2013) \log_2 \left(\frac{1}{0.2013}\right) + (0.1525) \log_2 \left(\frac{1}{0.1525}\right)$$
$$+ (0.1329) \log_2 \left(\frac{1}{0.1329}\right) + (0.1102) \log_2 \left(\frac{1}{0.1102}\right) + (0.1086) \log_2 \left(\frac{1}{0.1086}\right)$$
$$+ (0.0526) \log_2 \left(\frac{1}{0.0526}\right)$$

$H(x) = 2.683$ bits/symbol

$$\bar{L} = \sum_{i=0}^{i} P(x_i) L_i$$

$$= (2)(0.2419 + 0.2013) + (3)(0.1525 + 0.1329 + 0.1102) + (4)(0.1086 + 0.0526)$$

$$\bar{L} = 2.718$$

$$\eta = \frac{2.683}{2.718} \times 100\%$$

$$= 98.71\%$$

...N 3193 - Data Communications
...ong Quiz No. 1

PUP College of Engineering
AY 2018-2019 2nd Semester

Problem 3. What should be the Signal-to-Noise Ratio (in dB) in order to maintain a capacity of 100 Mbps at 20-MHz bandwidth? *(5 pts.)*

Problem 4. A communication system with 24-dB SNR is operating at 32-MHz bandwidth. What is the maximum bit rate that can be achieved on this system? *(5 pts.)*

Given:
SNR = 24dB
C = 32 MHz
required:
B = ?
Formula:
$C = B\log_2(1+\frac{S}{N})$
Solution:
$C = B\log_2(1+\frac{S}{N})$
$B = \frac{C}{\log_2(1+\frac{S}{N})}$
$B = \frac{32\ MHz}{\log_2(10^{\frac{24}{10}})}$
B = 4.01Mbps

Problem 5. Aling Florencia, a rich old woman, wants to communicate to his loved one in Siberia by using a high-speed dedicated radio teletype (RTTY) system. Applying from what she have learned during her college days, she decided to compare three communication systems. System A operates at 15-MHz bandwidth with nominal SNR of 24 dB, System B operates at 10-MHz bandwidth with nominal SNR of 41 dB, and System C operates at 8-MHz bandwidth with nominal SNR of 52 dB. If you will be Aling Florencia, what system would you choose? Show your justification to the answer. *(10 pts.)*

System A:
Given:
C = 15 MHz
SNR = 24dB
required = B = ?
formula:
$C = B\log_2(1+\frac{S}{N})$
Solution:
$B = \frac{C}{\log_2(1+\frac{S}{N})}$
$= \frac{15\ MHz}{\log_2(10^{\frac{24}{10}})}$
B = 1.886 Mbps

System B
Given:
C = 10 MHz
SNR = 41 dB
Req'd = B = ?
Formula:
$C = B\log_2(1+\frac{S}{N})$
Solution:
$B = \frac{C}{\log_2(1+\frac{S}{N})}$
$B = \frac{10\ MHz}{\log_2(10^{\frac{41}{10}})}$
B = 0.734 Mbps

System C:
Given:
C = 8 MHz
SNR = 52dB
Req'd = B = ?
Formula:
$C = B\log_2(1+\frac{S}{N})$
Solution:
$B = \frac{C}{\log_2(1+\frac{S}{N})}$
$B = \frac{8\ MHz}{\log_2(10^{\frac{52}{10}})}$
B = 0.963 Mbps

As shown System A is the best among the 3 systems.

5

CN 3193 - Data Communications
Long Quiz No. 1

PUP College of Engineering
AY 2018-2019 2nd Semester

BONUS QUESTIONS. This is not a required part. But for every question answered correctly, it will be added to your total score.

1. Cebu province is located at what region? *(2 pts.)*

   Region III ✓

2. Letter with Morse code equivalent of " — — — " *(2 pts.)*

   T ✓

3. *In vino veritas* in english means _____. *(5 pts.)*

   Good luck ✓

4. This bridge connects Solana town and Tuguegarao City in Cagayan. *(5 pts.)*

   ———

5. Which of the two became a city first: Marikina or Muntinlupa? *(2 pts.)*

   Marikina ✓

6. How many PUP campus are there in Mindoro Island? *(2 pts.)*

   0 ✓

7. It is the process where molecules tend to pass a semipermeable membrane from a less concentrated solution into a more concentrated one. *(5 pts.)*

   Permeability ✓

8. How many articles are there in the 1986 Philippine constitution? *(2 pts.)*

   24 ✓

9. In the 2018 Starbucks Planner promo, how many stickers should a customer have to obtain a planner? *(2 pts.)* 15 ✓

10. The epicenter of the 1990 Luzon earthquake was located at the province of _____. *(5 pts.)*

    Batangas ✓

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
STA. MESA, MANILA
COLLEGE OF ENGINEERING
COEN 3134 – LOGIC CIRCUITS AND SWITCHING THEORY
QUIZ #2

Name: _____ Score: _____

Student No.: _____ CYS: _____ Date: _____

GENERAL INSTRUCTIONS:

1. Follow all instructions carefully. Failure to do so will warrant a substantial deduction from your final score.
2. You are not allowed to leave your seat unless you are through with the exam. If you have any questions, just raise your hand and the instructor or proctor will attend to you.
3. Talking to or looking at your seatmate (and his her paper) is automatically considered as cheating which is subject to very serious sanctions as stipulated in the student handbook.
4. Cellphones, Notebooks and Calculators are not allowed during examination.
5. Use blue or black pen only. Erasable pens are not allowed. GOODLUCK!

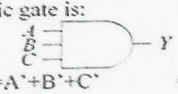
I. Write the letter of the correct answer before the number. If the answer is not among the choices, write **NULL**. Strictly no erasures and alterations. (30pts)

_____1. It is the mathematics of digital system.
a. Boolean Algebra b. Abstract Algebra c. Linear Algebra d. College Algebra

_____2. It is a small silicon semiconductor crystal (chip) containing electrical components such as transistors, diodes, resistors and capacitors.
a. Integrated Circuit b. Digital Trainer c. Protoboard d. IC Puller

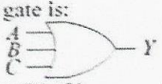
_____3. A logic function in which 1 output occurs only when the two inputs are at opposite levels.
a. NAND b. AND c. OR d. XOR

_____4. The Boolean expression for the logic gate is:



a. Y=A+B+C b. Y=A`+B`+C` c. Y=ABC d. Y=(ABC)`

_____5. The Boolean expression for the logic gate is:

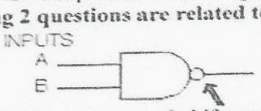


a. Y=ABC b. Y=A`+B`+C` c. Y=A+B+C d. Y=(A+B+C)`

_____6. When used with an IC, what does the term "QUAD" indicate?
a. 2 circuits b. 4 circuits c. 6 circuits d. 8 circuits

_____7. A Boolean equation in product-of-sum (POS) form is written in what format?
a. ANDed expression ANDed together b. ORed expression ORed together
c. ANDed expression ORed together d. ORed expression ANDed together

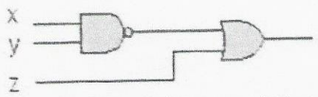
The following 2 questions are related to the image below:



_____8. What inputs are needed if output=0?
a. A=0, B=0 b. A=0, B=1 c. A=1, B=0 d. A=1, B=1

_____9. What is the output of the above gate if input A=0, B=0.
a. 1 b. A c. 0 d. B

The following 2 questions are related to the image below:



_____10. What is the output if input X=1, Y=0 and Z=1?
a. 1 b. X c. 0 d. Y

_____11. What inputs are needed if output=0?
a. X=1, Y=1, Z=0 b. X=0, Y=1, Z=0 c. X=1, Y=1, Z=1 d. X=0, Y=0, Z=1

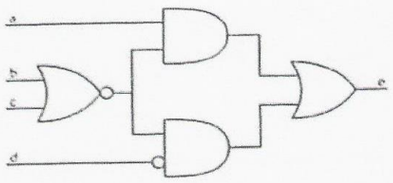
_____12. An Exclusive-NOR function is expressed as.
a. A`B + AB` b. A`B` + AB c. (A` + B)(A+B`) d. (A` + B`) + (A +B)

_____13. The logical expression $A + \bar{A}B$ is equivalent to
a. AB b. A+B c. A d. B

_____14. An equivalent representation for the Boolean expression A` + A is
a. A b. A` c. 1 d. 0

_____15. An equivalent representation for the Boolean expression A' + A is
a. A b. A' c. 1 d. 0

Given the Combinational Circuit



_____16. What is the Boolean equation of the given figure?
a. $E = A\overline{(B + C)} + \overline{DB + C}$ b. $E = (A + \overline{BC})(\overline{D} + \overline{BC})$
b. $E = A(\overline{B} + \overline{C}) + \overline{D}(\overline{B} + \overline{C})$ d. $E = (A + \overline{B}\overline{C})(\overline{D} + \overline{B}\overline{C})$

_____17. A TTL IC Package for NAND Gate?
a. 74LS02 b. 74LS00 c. 74LS86 d. 74LS266

_____18. A TTL IC Package for XNOR Gate?
a. 74LS02 b. 74LS00 c. 74LS86 d. 74LS266

_____19 A TTL IC Package for OR Gate?
a. 74LS02 b. 74LS00 c. 74LS86 d. 74LS266

_____20. A TTL IC Package for NOR Gate?
a. 74LS02 b. 74LS00 c. 74LS86 d. 74LS266

_____21. The expression A' + B' + C is represented by which of the following canonical symbols
a. m1 b. M1 c. m6 d. M6

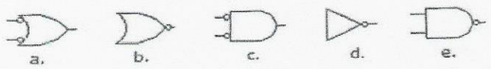
_____22. The Boolean function F(A,B,C) = $\prod$(0,1,3,5,6,7) is the same as
a. m2+m4 b. M2+m4 c. M2+M4 d. m2+M4

_____23. The Boolean function X (A,B) = Σ(1,2) is the same as
a. m1+m2 b. M1+m2
c. M1+M2 d. m1+M2

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

_____24. Which logic gate does this given truth table describe?
a. AND b. OR
c. NAND d. NOR

_____25. Which of the figures (a to d) is the DeMorgan equivalent of Figure (e)?



a. a b. b c. c d. d

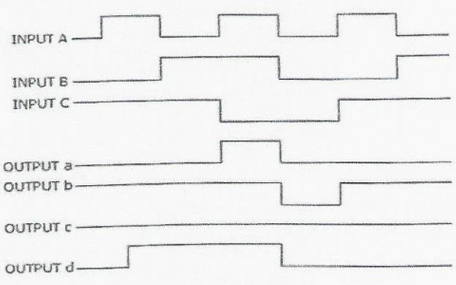
_____26. A small circle on the output of a logic gate is used to represent
a. OR operation b. AND operation c. NOT operation d. NOR operation

_____27. Simplify the expression $\overline{\overline{AB} + C}$ using DeMorgan's Theorem
a. $\overline{\overline{AB} + C}$ b. $AB\bar{C}$ c. $AB + \bar{C}$ d. $(A + B)\bar{C}$

_____28. The simplified form of a given logic function below is X=A'+AC' and X=AC'. This state is
a. True b. False, it is A'+C
c. False, it is C' d. Cannot be simplified

_____29. Which of the following boolean expression will be excluded after simplifying the equation F(X,Y,Z)=XY+YZ+X'Z using Consensus Theorem,
a. XY b. YZ c. X'Z d. Y'Z'

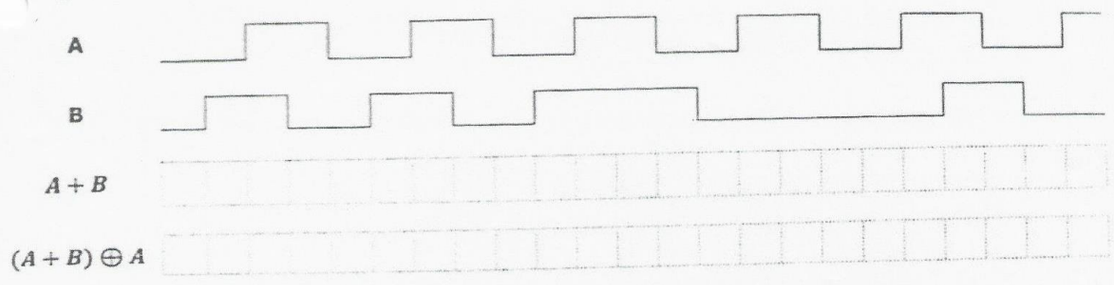
_____30. For a three-input OR gate, with the input waveforms as shown below, which output waveform is correct?



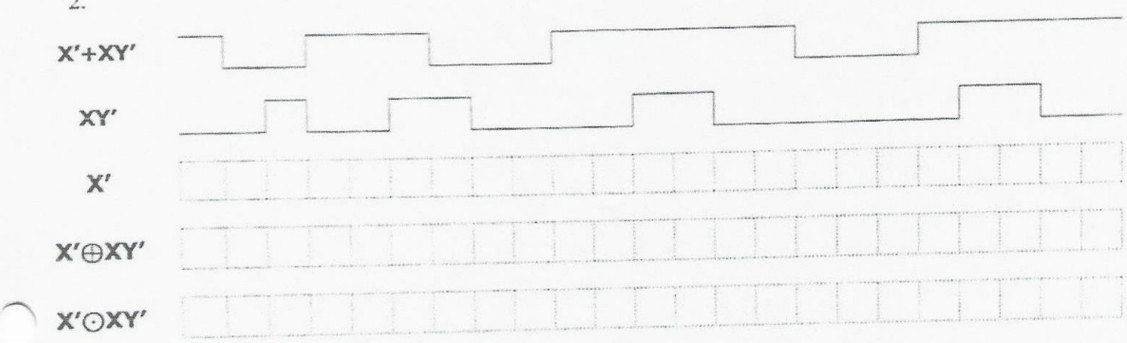
a. OUTPUT A b. OUTPUT B
c. OUTPUT C d. OUTPUT D

II. **WAVEFORM TRACING:** Construct the waveform behavior of the following logic gates given. Erasure means wrong (10pts)

1.



2.



III. Encircle the incorrect values in the truth table. RIGHT MINUS WRONG (1pt each)

Given: $F(A,B,C) = \bar{A}\bar{C} + [AB + \bar{B}][(AB + \bar{B}) \oplus C]$

A	B	C	$\bar{A}$	$\bar{B}$	$\bar{C}$	AB	$\bar{A}\bar{C}$	$AB + \bar{B}$	$(AB + \bar{B}) \oplus C$	F
0	0	0	1	1	1	0	1	1	1	0
0	0	1	1	1	0	0	0	1	0	1
0	1	0	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	1	0	1
1	0	1	0	1	0	0	0	1	0	0
1	1	0	0	0	1	1	1	0	1	1
1	1	1	0	0	0	1	0	1	0	0

IV. Answer the following problems. Show your solutions (25pts)

1. Simplify the following Boolean expressions:
   a. $F(A,B,C,D) = A + [AB'(C + BD) + A'B'']$[C]

b. $F(A, B, C) = \Pi(0, 2, 6)$

2. Construct the minterm notation of the Boolean equation $F(A, B, C, D) = AB' + BCD + ACD + A'BCD$

3. Given Boolean equation: $F(A, B, C) = (A \oplus B'C') + B'(A + C)$. Construct the combinational circuit diagram and its truth table.

===END OF TEST===

"A man is like a fraction whose numerator is what he is and whose denominator is what he thinks of himself, the larger the denominator, the smaller the fraction." -- Tolstoy

POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
STA. MESA, MANILA
COLLEGE OF ENGINEERING
CMPE 20012 – COMPUTER FUNDAMENTALS AND PROGRAMMING
QUIZ #2

Name: _____ Score: _____

Student No.: _____ CYS: _____ Date: _____

GENERAL INSTRUCTIONS:

1. Follow all instructions carefully. Failure to do so will warrant a substantial deduction from your final score.
2. You are not allowed to leave your seat unless you are through with the exam. If you have any questions, just raise your hand and the instructor or proctor will attend to you.
3. Talking to or looking at your seatmate (and his/her paper) is automatically considered as cheating which is subject to very serious sanctions as stipulated in the student handbook.
4. Cellphones, Notebooks and Calculators are not allowed during examination.
5. Use blue or black pen only. Erasable pens are not allowed. GOODLUCK!

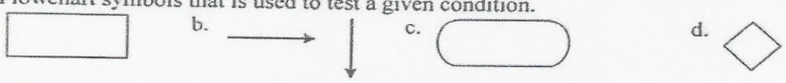
I. Write the letter of the correct answer before the number. If the answer is not among the choices, write **NULL**. Strictly no erasures and alterations. (25pts)

_____1. It is the graphical representation that represents algorithm.
a. Algorithm b. Pseudo Code c. Flow chart d. Programs

_____2. Which of the following symbols in evaluating the remainder of two variables in Visual Logic.
a. / b. % c. r d. MOD

_____3. It is a preprocessor instruction that is used to include certain files into the program.
a. #define b. #include c. main() d. using namespace std

_____4. Flowchart symbols that is used to test a given condition.
a. b. c. d.



_____5. A preprocessor instruction that is used to declare a macro constants.
a. #define b. #include c. main() d. using namespace std

_____6. Consider the following code fragment carefully, then answer the question: how many times will the cout statement execute:

```
for (i = 0; i < 5; i++);
cout << i;
```

a. 5 times b. 4 times c. 6 times d. 0 times

_____7. The end-of-output in visual logic that always appears at the end of the console output expression.
a. § b. ; c. & d. EXIT LOOP

_____8. This is used for blocking the statements.
a. {} b. () c. [] d. <>

_____9. A symbol used for statement terminator.
a. colon b. semi-colon c. period d. white space

_____10. Input stream object symbol.
a. << b. >> c. \ d. #

_____11. Assigns value to a variable.
a. arithmetic operator b. binary operator c. assignment operator d. relational operator

_____12. What will be the output of the given expression (5+20/10*3)?
a. 18 b. 12 c. 15 d. 42

_____13. Set of statements with in a braces.
a. condition b. pre-processor c. looping d. block

_____14. What is the final value of x when the code `int x; for (x=0; x<=10; x++) {}` is run?
a. 10 b. 9 c. 0 d. 11

_____15. post-decrement value of y.
a. --y b. y-- c. y-+ d. y+-

_____16. x+=y is the same as
a. x=+y b. x==+y c. x=x+y d. y+=x

_____17. Suppose that num1 is an int variable and the input is: 25
What is the value of num1 if the expression is num1+=25.
a. 0 b. 25 c. 50 d. 250

_____18. Symbol used for single line comment.
a. /* */ b. // c. ## d. \\

_____19. Suppose that x and y are int variables and the input is: `10 20.7`
What is the value of x and y if the inputs are already executed: cout << x << y;
a. x=10, y=20 b. x=1, y=2 c. x=10, y=20.7 d. x=10, y=21

_____20. Which of the following is the "not equal" operator in Visual Logic.
a. ! b. | c. & d. !=

_____21. Suppose that x is an int variable. Which of the following expressions always evaluates to true?
a. (x > 0) || (x <= 0) b. (x >= 0) || (x == 0)
c. (x > 0) && (x <= 0) d. (x > 0) && (x == 0)

_____22. It is a generalization of the traditional counted loop that appears in most programming languages.
a. for statement b. if/else statement c. do-while statement d. switch statement

_____23. It is like a while statement, except that it tests at the end of the loop body.
a. for statement b. if/else statement c. do-while statement d. switch statement

_____24. If a and b are both integer data type and a=3, b=2. Evaluate a^b.
a. 0 b. 1 c. 6 d. 9

_____25. A data type that can be assigned the literal values True or False.
a. int b. double c. bool d. char

II. PROGRAM EXPRESSIONS: Write the exact output of the given program code. (15pts)

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,c;
    a=1; b=4;

    for(int i=0;i<=8;i++)
    {
        if(i==1)
        continue;

        switch(i)
        {
            case 2:
                cout<<a++;
            case 4:
            {
                cout<<++a<<endl;
                break;
            }
            default:
                cout<<a;
        }
        c = ++b - a++;
        if(i==5)
        break;
    }
    return 0;
}
```

Answer:

```
#include <iostream>
using namespace std;
int main()
{
      int x=5, y=10, z=6;

      cout << (y & z) + (x | y) << endl;
      cout << (z>>2)*(y<<3)/(x + 6) << endl;

      y = ++x + z--;
      cout << "x: " << x << endl;
      cout << "y: " << y++ << endl;
      cout << "z: " << ++z <<endl;

      x+=5; y%=2;   z-=x; z*=-1;
      cout << "x: " << x << endl;
      cout << "y: " << y-- << endl;
      cout << "z: " << ++z <<endl;
      return 0;
}
```

Answer:

III. **PROGRAM CODING**: Encircle the errors in the source code. (5pts)

Source Code:

```
#include <iostream>
using namespace std;
int main()
{
       integer x,y;
       cout >> "Enter two numbers: ";
       cin >> x >> y;
       if(x<y)
       {
       cout << x << " is greater than " << y;
       }
       elseif(x<y)
       {
       cout << y << " is greater than " << x;
       }
       else
       {
       cout << y << " is equal to " << x;
       }

      return;
}
```

OUTPUT 1:

```
Enter two numbers: 8 4
8 is greater than 4
```

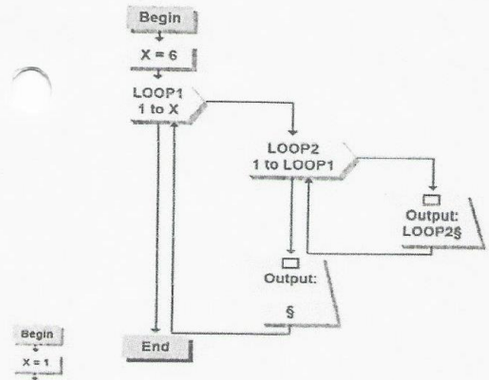
OUTPUT 2:

```
Enter two numbers: 10 30
30 is greater than 10
```

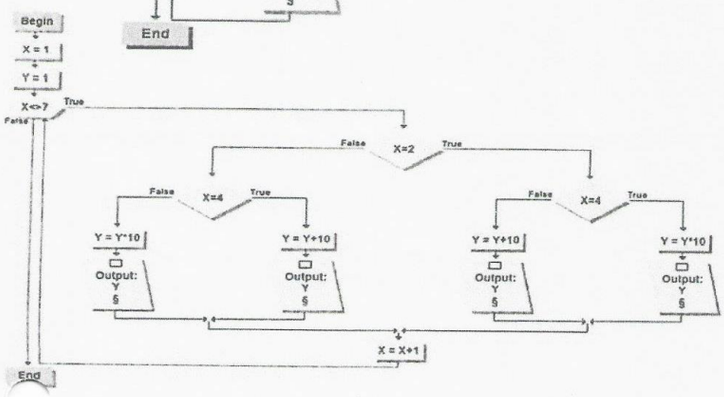
OUTPUT 3:

```
Enter two numbers: 5 5
5 is equal to 5
```

IV. **FLOWCHART**: Write the exact output of the given flowchart. (15pts)



Answer:



Answer:

==END OF TEST===

"The Only Reward in Telling the TRUTH is the GOOD FEELING It leaves behind"

RANARIO, RAFAEL L.
BSCOE 1-4
COMPENG TECH 1

SEMICONDUCTOR CODING

Electronic Industries of Japan, USA, and Europe use different codes of their semiconductor products. Here are some examples of these codes.

1. **Japanese Semiconductor** – standards of type-number-meaning are set by the Electronics Industries Association of Japan (EIAJ).

2	S	C	1451	A
1st	2nd	3rd	4th	5th

1st (number) – represents number of effective junctions

0	Phototransistor or Photodiodes
1	Diodes
2	Bipolar transistor, FET, SCR, TRIAC, etc.
3	Four-lead transistor: MOSFET

2nd (character) – 'S' represents semiconductor

3rd (character) – represents types and uses, if it is E or I it is used as a diode.

A	PNP high frequency transistor
B	PNP low frequency transistor
C	NPN high frequency transistor
D	NPN low frequency transistor
F	P-type gate SCR
G	N-type gate SCR
H	UJT (unijunction transistor)
J	P-channel FET
K	N-channel FET
M	Triac

4th (number) – represents EAIJ register number

5th (character) – represents the improved version of the original type; it uses A,B,C,etc.

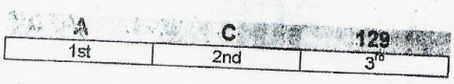
2. **U.S. Semiconductor** – standards are set by the Joint Electron Device Council (JEDEC).

1st (alphanumeric) – represents number of leads and semiconductor types

1N	Two-lead diodes
2N	Three-lead transistors, UJTs, thyristors, etc.
SK	Amateur's or hobbyist's semiconductor (RCA)
MJE	Plastic Power Transistor (Motorola)
HEP	Hobbyist's and experimenters

2nd (number) – represents JEDEC register number

3. **European Semiconductors**



1st (character) – represents semiconductor material used.

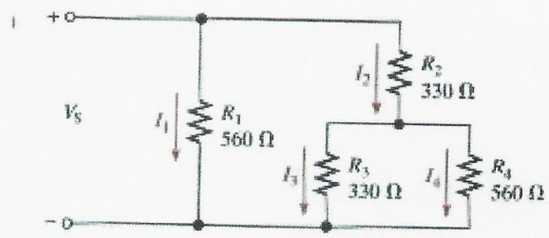
A	Germanium
B	Silicon
C	Gallium Arsenide
D	Indium Antimonide
R	Hall Effect or Photovoltaic material

2nd (character) – represents type and applications

A	Detector, switching or mixing diode
B	Varactor diode
C	Audio frequency transistor
D	Audio frequency power transistor
E	Tunnel diode
F	High frequency transistor
H	Magnetic sensitive transistor
L	High frequency power transistor
S	Switching transistor
U	Switching power transistor
Y	Rectifying diode
Z	Zener diode

3rd (character) – represents nature of application; if all are numeric such as AC187, the product is used on ordinary or common instrument; if the first is an alpha character (letter) followed by two numeric, such as BXY27, it is used in special or industrial equipment.

3. Determine the current through R4, if Vs = 50 V.



Let

$R_A = R_3 + R_4$

$R_B = R_2 + R_A$

$R_T = R_1 + R_B$

$R_A = R_3 + R_4$

$R_A = \frac{1}{\frac{1}{330} + \frac{1}{560}}\ \Omega$

① $R_A = 207.640\ \Omega$

$R_B = 330 + 207.640\ \Omega$

② $R_B = 537.640\ \Omega$

$1/R_T = 1/560 + 1/537.64\ \Omega$

③ $R_T = 274.296\ \Omega$

$I_T = \frac{V_T}{R_T}$

$I_T = \frac{50V}{274.296\Omega}$

④ $I_T = 0.182A / 0.18A$

$I_1 = \frac{50V}{560\Omega}$

$I_1 = 0.09A.$

$I_B = \frac{50V}{537.64\Omega}$

$I_B = 0.09A.$

$V_A = 0.09A\ (207.64)$

$V_A = 18.69V$

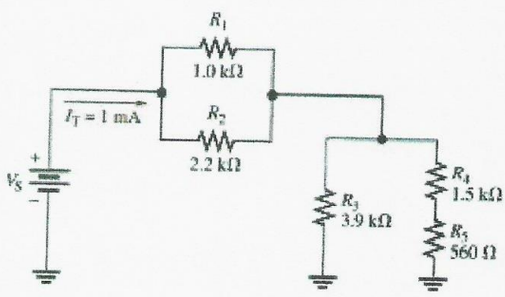
$I_3 = \frac{18.69V}{330\Omega}$

$I_3 = 0.06A$

$I_4 = \frac{18.09V}{560\Omega}$

$I_4 = 0.03A.$

4. Determine the voltage across each resistor in the circuit?



Let

$R_A = R_1 + R_2$

$R_B = R_4 + R_5$

$R_C = R_3 + R_B$

$R_T = R_A + R_C$

$\frac{1}{R_A} = \frac{1}{1000\Omega} + \frac{1}{2200\Omega}$

$R_A = 687.5\ \Omega$

$R_B = 1500\Omega + 560\Omega$

$R_B = 2060\Omega$

$1/R_C = 1/2060 + 1/3900$

$R_C = 1347.99\ \Omega$

$R_T = 687.5\Omega + 1347.99\Omega$

$R_T = 2035.49\ \Omega$

$V_A = 687.5\Omega \cdot 0.001A$

$= 0.69V$

$V_C = 1347.99\Omega \cdot 0.001A$

$= 1.35V$

$I_3 = \frac{1.35V}{3.9k\Omega}$

$I_3 = 3.46 \times 10^{-4} A.$

$I_B = \frac{1.35V}{2060\Omega}$

$I_B = 6.55 \times 10^{-4} A.$

$V_4 = 0.983V$

$V_5 = 0.367V$

$I_1 = 0.69mA$

$I_2 = .314\ mA$

$V_1 = .69V$

$V_2 = .69V$

VOLTAGES:

$V_1 = .69V$

$V_2 = .69V$

$V_3 = 1.35V$

$V_4 = 0.983V$

$V_5 = 0.367V$

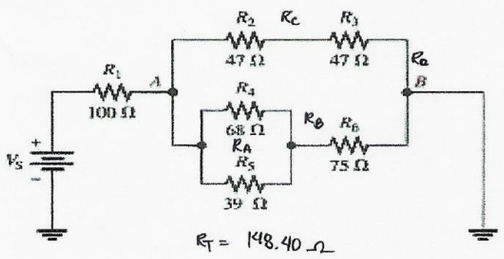
POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
College of Engineering
Department of Computer Engineering
Sta. Mesa, Manila

First Semester, AY 2019-2020

Name RANARIO, RAFAEL L.	**Section** BSCOE 1-4	**Date** AUGUST 17, 2019

Homework # 3

1. Find the total resistance in the circuit.



$R_T = 148.40\ \Omega$

SOL'N:

Let: $R_A = R_4 + R_5$ $\quad R_D = R_B + R_C$

$R_B = R_A + R_a$ $\quad R_T = R_D + R_1$

$R_C = R_2 + R_3$

$1/R_A = 1/_{68} + 1/_{39}$

① $R_A = 24.79\ \Omega$

$R_B = 24.79\ \Omega + 75\ \Omega$

② $R_B = 99.79\ \Omega$

$R_C = 47\ \Omega + 47\ \Omega$

③ $R_C = 94\ \Omega$

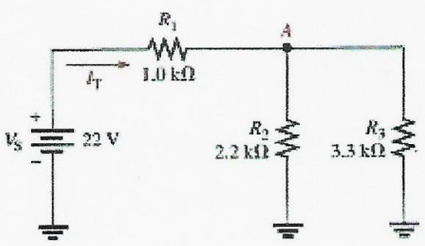
$1/R_D = 1/99.79\ \Omega + 1/94\ \Omega$

④ $R_D = 48.40\ \Omega$

$R_T = 48.40\ \Omega + 100\ \Omega$

⑤ $R_T = 148.40\ \Omega$

2. Find the current through R2 and the current through R3.



SOL'N:

$R_A = R_2 + R_3$

$R_T = R_A + R_1$

$\frac{1}{R_A} = \frac{1}{8200} + \frac{1}{3300}$

$R_A = 1320\ \Omega$

$R_T = 1320\ \Omega + 1000\ \Omega$

$R_T = 8320\ \Omega$

① $I_T = \frac{8.2V}{8320\ \Omega}$

$I_T = 9.48 \times 10^{-3}\ A$

$I_1 = I_T = 0.00948A$

② $V_A = R_A \cdot I_1$

$V_A = 1320\ \Omega \cdot 0.00948A$

$V_A = 12.51V$

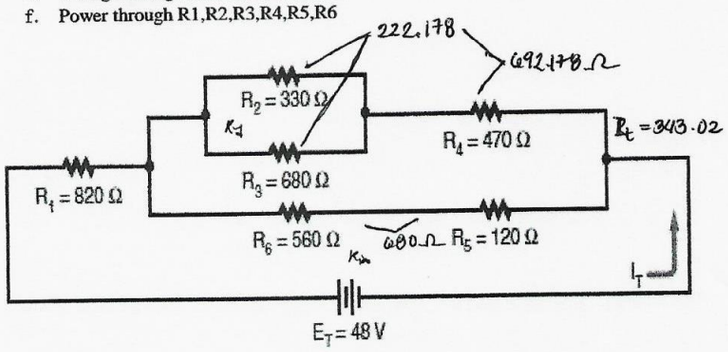
③ $I_2 = \frac{12.51V}{8200\ \Omega}$

$= 5.688\ mA$

④ $I_3 = \frac{12.51V}{3300\ \Omega}$

$I_3 = 0.99mA$

5. Solve for all unknown quantities in the circuit
   a. Total current 0.0412 A
   b. Total resistance 1163.01 Ω
   c. Total power 1.98 W
   d. Current through R1,R2,R3,R4,R5,R6
   e. Voltage through R1,R2,R3,R4,R5,R6
   f. Power through R1,R2,R3,R4,R5,R6



$I_1 = 99.07\ A$
$I_2 = 19.97\ A$
$I_3 = 19.97\ A$
$I_4 = 39.95\ A$
$I_5 = 59.27\ A$
$I_6 = 178.27\ mA$

$R_t = 1163.01\ \Omega$

$I_t = 0.04127\ A$

$R_1 = 820\ \Omega$ $\quad I_1 = 0.04127\ A$ $\quad V_1 = 33.8414\ V$

$R_{tp} = 343.02\ \Omega$ $\quad I_{TP} = 0.04127\ A$ $\quad V_{TP} = 14.156\ V$

$V_1 = 100.6\ mV$
$V_2 = 20.09\ mV$
$V_3 = 20.09\ mV$
$V_4 = 40.10\ mV$
$V_5 = 59.88\ mV$
$V_6 = 297.67\ \mu V$

$R_A = R_2 + R_3 = \frac{1}{1/330 + 1/680} = 222.178\ \Omega$

$R_B = R_5 + R_6 = 560\ \Omega + 120\ \Omega = 680\ \Omega$

$R_C = R_A + R_4 = 692.178\ \Omega$

$R_D = R_B + R_C = \frac{1}{1/680 + 1/692.178} = 343.02\ \Omega$

$R_t = R_1 + R_D = 820 + 343.02$

$R_T = 1163.02\ \Omega$

$I_T = \frac{48\ V}{1163.02\ \Omega} = 0.041\ A$

$P_T = I^2R = (0.041\ A)^2\ 1163.02\ \Omega = 1.98\ W$

$I_1 = 0.59\ A$

$V_1 = 33.62\ V$ $\quad V_A = 4.44\ V$
$V_D = 13.702\ V$ $\quad V_4 = 19.34\ V$
$V_C = 13.84\ V$ $\quad V_2 = 4.44\ V$
$V_B = 13.6\ V$ $\quad V_3 = 4.44\ V$
$V_6 = 11.2\ V$
$V_5 = 2.4\ V$

$P_1 = 1.31\ W$
$P_2 = 0.04\ W$
$P_3 = 0.04\ W$
$P_4 = 0.19\ W$
$P_5 = 0.05\ W$
$P_6 = 0.22\ W$

TOPIC: LINKED LIST

Introduction

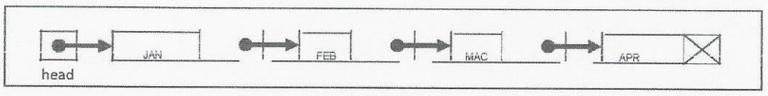
In computer science, a **list** or **sequence** is an abstract data structure that implements an ordered collection of values, where the same value may occur more than once. Each instance of a value is often called an item, entry, or element of the list; if the same value occurs multiple times, each occurrence is considered a distinct item.

List is a collection of data, element, component or objects with similar data type. List always represented as a record. Generally, a collection of data items that can be selected by indices computed at run-time, including array data structure, an arrangement of items at equally spaced addresses in computer memory. List can be implemented using array that contains sequence of data/record.

listArray	12	14	16	18	20
	[0]	[1]	[2]	[3]	[4]

A list of number using array

In computer science, a **linked list** is data structure that consists of a sequence of data records such that in each record there is a field that contains a reference (i.e., a *link*) to the next record in the sequence.



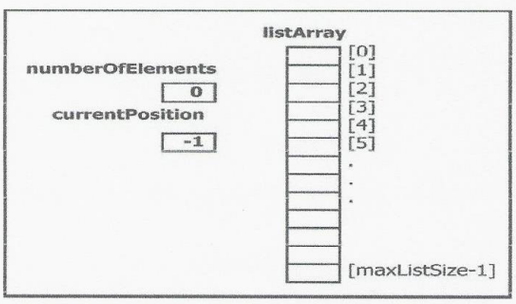
A linked list whose nodes contain two fields: a string value and a link to the next node

The principal benefit of a linked list over a conventional array is that the order of the linked items may be different from the order that the data items are stored in memory or on disk. For that reason, linked lists allow insertion and removal of nodes at any point in the list, with a constant number of operations.

Concept

List Characteristic

- maximum size
- array for storing entries
- number of elements/entries
- current position



(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

List Operations

Operations involved in implementing list using array are createList, insertItem, and DeleteItem. For createList operation, just declare an array.

Example, if want to create list of 10 number, declared as

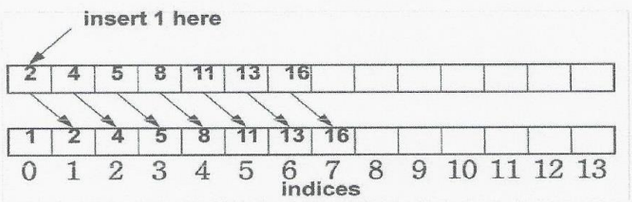
```
#define maxSize 10          // used to declare maximum size of array
int listNumber[maxSize];    // declaration of an array
int NoOfItem;               // used to store no. of item in an array
```

To verify whether an array are empty or not, just check the `NoOfItem` variable. If `NoOfItem` is equal to zero (0), it shows that the list is empty. Looping statement can be used to traverse the list of array; that loop accordingly to the `NoOfItem` in the list. For insertion and deletion operation, the implementation is more complicated based on the data type of the list.

Problem with array

- Array implementations of lists use a static data structure. Often defined at compile-time. Cannot be altered while program is running.
- This means we usually waste space rather than have program run out.
- It also means that it is difficult to construct ordered lists. In our implementation, data must be added to the end. If inserted before the end, all others beneath it must shuffle down. This is slow and inefficient.

Insertion shuffle



Limitation of array

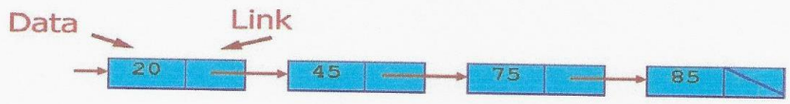
- An array has a limited number of elements
  - routines inserting a new value have to check that there is room
- Can partially solve this problem by reallocating the array as needed (how much memory to add?)
  - adding one element at a time could be costly
  - one approach - double the current size of the array
- A better approach: use a Linked List
  - and dynamically allocate memory

Why Linked Lists?

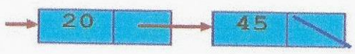
Linked lists and arrays are similar since they both store collections of data. The terminology is that arrays and linked lists store "elements" on behalf of "client" code. The specific type of element is not important since essentially the same structure works to store elements of any type. One way to think about linked lists is to look at how arrays work and think about alternate approaches.

Linked List: Basic Ideas

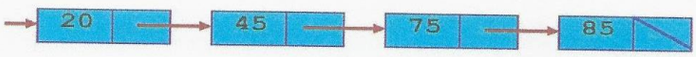
- A linked list is an *ordered* series of connected data / nodes
- Each element of the linked list has
  - *Some data*
  - *A link to the next element*
- The link is used to chain the data
- Example : A linked list of integers



- The linked list can grow and shrink



add(75), add(85)



delete(85), delete(45), delete(20)

Linked List Structure

Before writing the code to build the above list, we need two data types...

- ***Node*** The type for the nodes which will make up the body of the list. Each node contains a single client data element and a pointer to the next node in the list. Type: `struct node`

```
struct node
{
      int data;
      struct node* next;
};
```

- ***Node Pointer*** The type for pointers to nodes. This will be the type of the head pointer and the `.next` fields inside each node. In C and C++, no separate type declaration is required since the pointer type is just the node type followed by a '*'. Type: `struct node*`

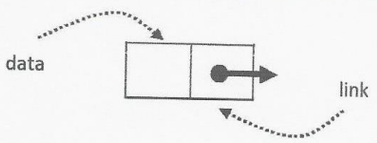
(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

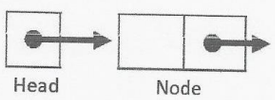
Data Structures and Algorithm

Memory Drawing

The best way to design and think about linked list code is to use a drawing to see how the pointer operations are setting up memory.

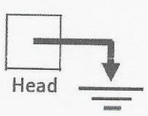
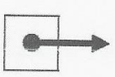
Code segment	Memory Drawing

• Declare structure for a node
`struct Node`
`{`
`int data;`
`Node *next;`
`};`
	

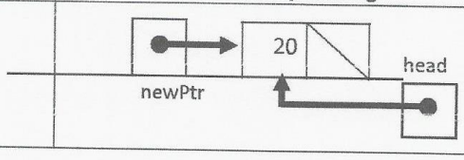
• Declare pointer
`typedef Node *NodePtr;`
`NodePtr Head;`
that points to the first node of the linked list. When the linked list is empty then Head is NULL.
	

Manipulation / Operation of Linked List

- Start the first node from scratch

Code segment	Memory Drawing
• Set the Head pointer to NULL value `Head = NULL;`	
• Create new pointer to pointer to a new node `NodePtr newPtr;`	

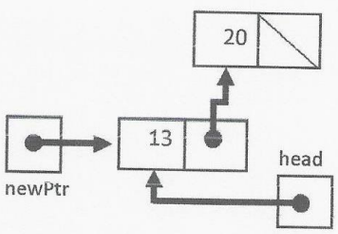
Code segment	Memory Drawing

• Create new node
`newPtr = new Node;`
`newPtr->data = 20;`
`newPtr->next = NULL;`
`Head = newPtr;`
	

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

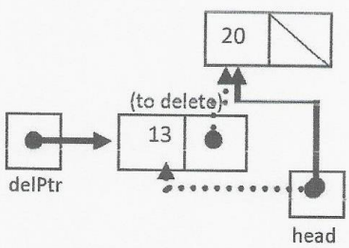
- Inserting / adding new node

Code segment	Memory Drawing

• Inserting a node at the beginning
`newPtr = new Node;`
`newPtr->data = 13;`
`newPtr->next = Head;`
`Head = newPtr;`
	

- Deleting / removing node

Code segment	Memory Drawing

• Deleting a head node
`NodePtr delPtr;`
`delPtr = head;`
`Head = Head->next;`
`Delete delPtr;`
	

Effectiveness of Linked List

Advantages	Disadvantages
Dynamically in size	Complex programming processed
Insert and delete operation doesn't need to shuffle an existing item	High in computerizing time and memory management
Less time used in insert and delete operation	Not suitable for simple list with minimal size
Used for large size of element (unknown size)	Complex in updating of program

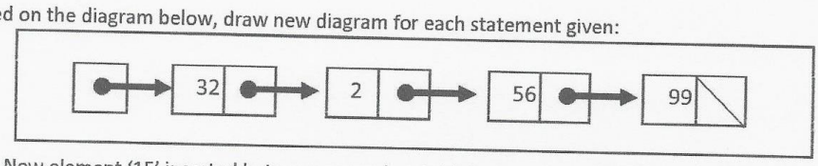
Differences between list and linked list

List	Linked List
• Static size of list	• Dynamic size of list
• Add & delete item required much steps	• Add & delete item required less steps
• Suitable for less of list	• Suitable for larger of list
• Easy programming processes	• Complex programming processes
• Easy program update	• Complicated program update
• Low of memory and computerizing time	• High of memory and computerizing time

Data Structures and Algorithm

Activity

1. Based on the diagram below, draw new diagram for each statement given:



   a) New element '15' inserted between second and third element in linked list
   b) Element 2 removed from the linked list

2. Illustrate deleting operation from linked list

Assessment

Q1. Linked list is

   A. A data structure that consists of a sequence of data records such that in each record there is a field that contains a reference
   B. An arrangement of items at equally spaced addresses in computer memory
   C. An abstract data structure that implements an ordered collection of values, where the same value may occur more than once
   D. A data structure that consists of a parallels of data records such that in each record there is a field that contains a reference

Q2. Linked list consists of at least

   A. Two fields : Data, link
   B. Two fields : Number, pointer
   C. Two fields : Integer, link
   D. Two fields : Number, link

Q3. Data structure for a node in linked list?

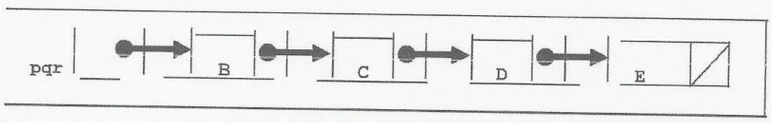
A.
```
struct Node
{
   int data;
   int *next;
}
```

B.
```
struct Node
{
   char data;
   Node *link;
}
```

C.
```
struct Node
{
    int data;
    Node next;
}
```

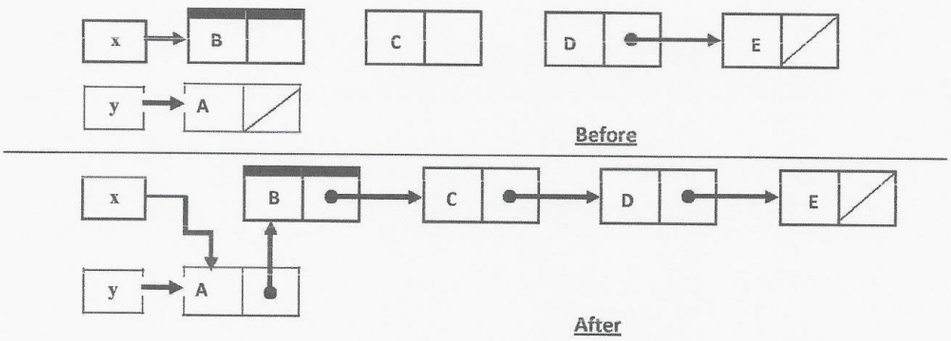
D.
```
struct Node
{
   char data;
   char *link;
}
```

Q4. Based on the diagram below, draw the changes that happen after `pqr = pqr->next->next` statement being executed:



(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Q5. Write down code statement to show an operation for inserting node in-front of the linked list based on the diagram given.



Q6. TRUE or FALSE?

A. In a linked list, components are only logically next to each other whereas in an array they are also physically next to each other.

B. Nodes in a linked list structure must contain a link member.

C. In deleting an item from a linked list, we need to keep track of the previous node.

Summary

- A data structure in which each element contains a pointer to the next element, thus forming a linear list.
- Linked lists are a way to store data with structures so that the programmer can automatically create a new place to store data whenever necessary.
- The linked list is relocatable, meaning it can be moved about in memory at will, and it can also be quickly and directly serialized for storage on disk or transfer over a network.
- A linked list is a dynamic data structure and therefore the size of the linked list can grow or shrink in size during execution of the program. A linked list does not require any extra space therefore it does not waste extra memory. it provides flexibility in rearranging the items efficiently.
- The limitation of linked list is that it consumes extra space when compared to a array since each node must also contain the address of the next item in the list to search for a single item in a linked list is cumbersome and time consuming.

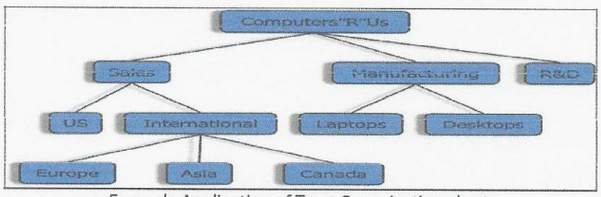
(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

TOPIC: BINARY TREE

Introduction

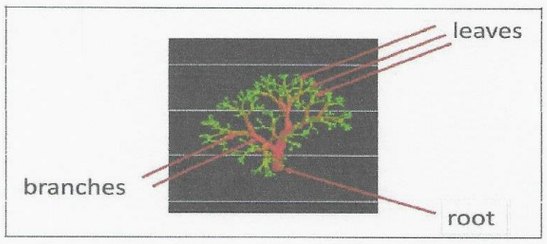
A **binary tree** is a tree data structure in which each node has at most two children. Typically the child nodes are called *left* and *right*.

- A tree is a finite non-empty set of elements.
- In computer science, a tree is an abstract model of a hierarchical structure
- A tree consists of nodes with a parent-child relation
- Applications:
  - Organization charts
  - File systems
  - Programming environments

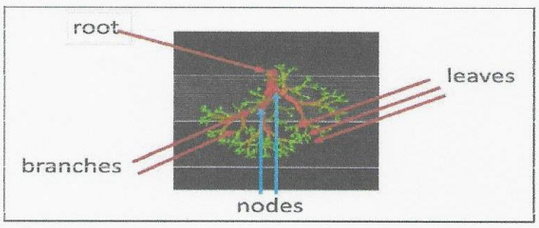


Example Application of Tree: Organization charts

Nature view of a Tree



Computer Science's view



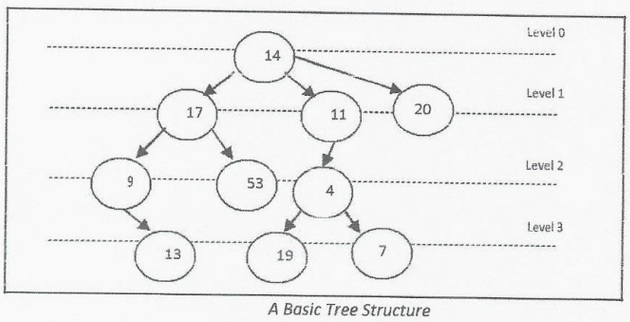
(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Data Structures and Algorithm

Concept

A *tree* is another data structure that you can use to store information. Unlike *stacks* and *queues*, which are linear data structures, trees are *hierarchical* data structures. Saying that the structure of a tree is *hierarchical* means that things are ordered *above* or *below* other things. For example, the army is hierarchical, with generals above colonels, and colonels above lieutenants, etc.

Here is an example of a tree holding integer numbers:



A Basic Tree Structure

Tree Terminology

Name	Explanation	Example
Root Node	Node without parent	14
External Node (Leaf Node)	Node without children	20, 53, 13, 19, 7,
Internal Node	Node with at least one child	14, 17, 11, 9, 4
Siblings Node	Node share the same parent	{17, 11, 20}, {9, 53}, {19, 7}
Height of tree	Maximum depth of any node	3
Degree of Node	The number of children for each node	Degree of node 14=3 Degree of node 17=2 Degree of node 11=1
Degree of Tree	The maximum degree of node in a tree	3
Sub-tree	tree consisting of a node and its descendants	

Binary Tree

A binary tree is made of nodes, where each node contains a "left" pointer, a "right" pointer, and a data element. The "root" pointer points to the topmost node in the tree. The left and right pointers recursively point to smaller "subtrees" on either side. A null pointer represents a binary tree with no elements -- the empty tree. The formal recursive definition is: a binary tree is either empty (represented by a null pointer), or is made of a single node, where the left and right pointers (recursive definition ahead) each point to a binary tree.

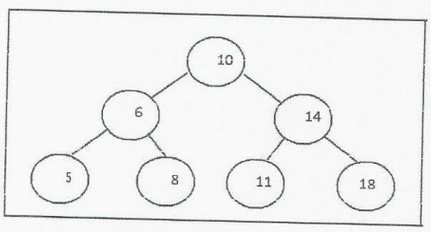
The binary tree is a fundamental data structure used in computer science. The binary tree is a useful data structure for rapidly storing sorted data and rapidly retrieving stored data. A binary tree is composed of parent nodes, or leaves, each of which stores data and also links to up to two other child nodes (leaves) which can be visualized spatially as below the first node with one placed to the left and with one placed to the right.

It is the relationship between the leaves linked to and the linking leaf, also known as the parent node, which makes the binary tree such an efficient data structure. It is the leaf on the left which has a lesser key

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

value (ie, the value used to search for a leaf in the tree), and it is the leaf on the right which has an equal or greater key value. As a result, the leaves on the farthest left of the tree have the lowest values, whereas the leaves on the right of the tree have the greatest values. More importantly, as each leaf connects to two other leaves, it is the beginning of a new, smaller, binary tree. Due to this nature, it is possible to easily access and insert data in a binary tree using search and insert functions recursively called on successive leaves.

The typical graphical representation of a binary tree is essentially that of an upside down tree. It begins with a root node, which contains the original key value. The root node has two child nodes; each child node might have its own child nodes. Ideally, the tree would be structured so that it is a perfectly balanced tree, with each node having the same number of child nodes to its left and to its right. A perfectly balanced tree allows for the fastest average insertion of data or retrieval of data. The worst case scenario is a tree in which each node only has one child node, so it becomes as if it were a linked list in terms of speed. The typical representation of a binary tree looks like the following:



A tree whose elements have at most *2 children is called a* binary *tree*

The node storing the 10, represented here merely as 10, is the **root node**, linking to the left and right child nodes, with the left node storing a lower value than the parent node, and the node on the right storing a greater value than the parent node. Notice that if one removed the root node and the right child nodes, that the node storing the value 6 would be the equivalent a new, smaller, binary tree.

The structure of a binary tree makes the insertion and search functions simple to implement using recursion. In fact, the two insertions and search functions are also both very similar. To insert data into a binary tree involves a function searching for an unused node in the proper position in the tree in which to insert the key value. The insert function is generally a recursive function that continues moving down the levels of a binary tree until there is an unused leaf in a position which follows the rules of placing nodes. The rules are that a lower value should be to the left of the node, and a greater or equal value should be to the right.

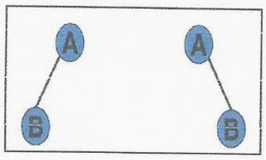
Following the rules, an insert function should check each node to see if it is empty, if so, it would insert the data to be stored along with the key value (in most implementations, an empty node will simply be a NULL pointer from a parent node, so the function would also have to create the node). If the node is filled already, the insert function should check to see if the key value to be inserted is less than the key value of the current node, and if so, the insert function should be recursively called on the left child node, or if the key value to be inserted is greater than or equal to the key value of the current node the insert function should be recursively called on the right child node.

The search function works along a similar fashion. It should check to see if the key value of the current node is the value to be searched. If not, it should check to see if the value to be searched for is less than the value of the node, in which case it should be recursively called on the left child node, or if it is greater than the value of the node, it should be recursively called on the right child node. Of course, it is also necessary to check to ensure that the left or right child node actually exists before calling the function on the node.

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

- A binary tree is a tree with the following properties:
  - Each internal node has at most two children (degree of two)
  - The children of a node are an ordered pair
- We call the children of an internal node left child and right child
- Alternative recursive definition: a binary tree is either
  - a tree consisting of a single node, OR
  - a tree whose root has an ordered pair of children, each of which is a binary tree
- Applications:
  - arithmetic expressions
  - decision processes
  - searching

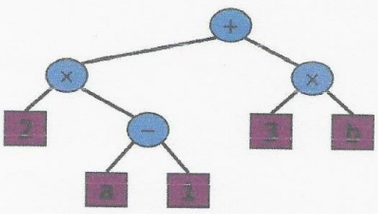
Differences between Tree and Binary Tree



- The sub trees of a binary tree are ordered; those of a tree are not ordered.
- Are different when viewed as binary trees.
- Are the same when viewed as trees.

Binary Tree Application: Arithmetic Expressions Tree

- Binary tree associated with an arithmetic expression
  - *internal nodes: operators*
  - *external nodes: operands*
- Example: arithmetic expression tree for the expression $(2 \times (a - 1) + (3 \times b))$
- Arithmetic operations should be performed in the following order:
  - parentheses
  - exponentiation
  - multiplication and division, left to right
  - addition and subtraction, left to right.



An arithmetic expression tree for (2x(a-1)+(3xb))

Data Structures and Algorithm

Activity 1

1. Based on figure (a) and (b), find root node, child node, leaf node, siblings' node, level, height and degree of the tree

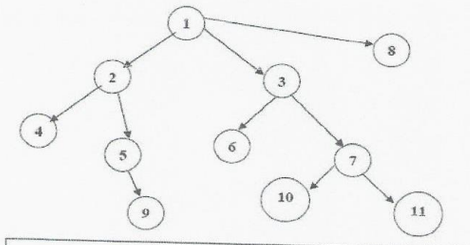


Figure (a)

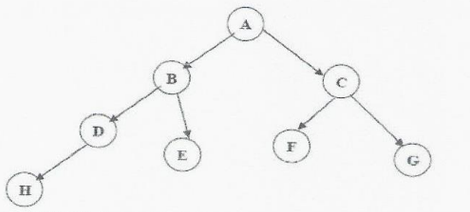


Figure (b)

2. Draw an arithmetic expression tree for all the statements below:

a) $(5 - x) * y + 6 / (x + z)$

b) $(3 + 4 * (6 - 7) / 5) + 3$

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

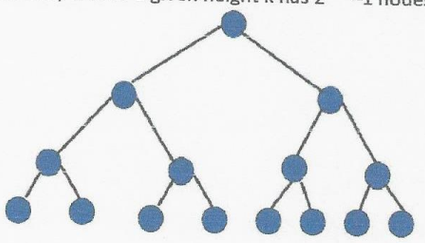
Maximum Number of Nodes in a Binary Tree

- The maximum number of nodes on depth i of a binary tree is 2^i, i>=0.
- The maximum number of nodes in a binary tree of height k is 2^{k+1}-1, k>=0.

Prove by induction.

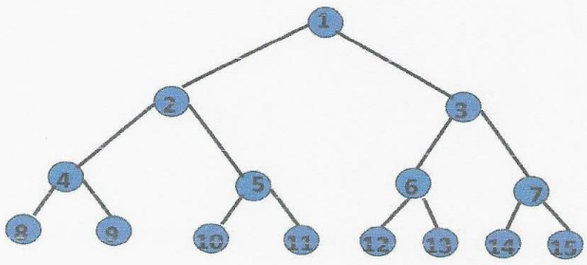
$$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$$

- Full Binary Tree - A full binary tree of a given height k has 2^{k+1}–1 nodes



Height 3 full binary tree.

- Labeling Nodes In A Full Binary Tree
  - Label the nodes 1 through $2^{k+1} - 1$.
  - Label by levels from top to bottom.
  - Within a level, label from left to right.



Activity 2

Draw a binary tree for :

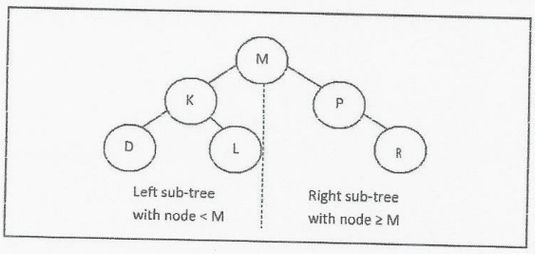
1. 20, 30, 45, 31, 23, 19, 15, 18, 13, 50, 21
2. M, O, R, T, C, F, E, A, S, N, Q

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Binary Search Tree

In computer science, a **binary search tree (BST)** is a binary tree data structure which has the following properties:

- Each node (item in the tree) has a distinct value.
- Both the left and right subtrees must also be binary search trees.
- The left subtree of a node contains only values lesser than the node's value.
- The right subtree of a node contains only values greater than the node's value.

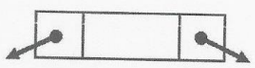


Implementation of Binary Search Tree

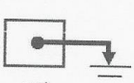
Node is an element of Binary Search Tree (BST) to store data or information. Each node in BST contained at least three fields; data field, left pointer field to point to the left node and right pointer field to point to the right node for BST.

- Example of data structure declaration for Binary Search Tree.

Code segment	Memory Drawing

• Declare structure of node for BST
`struct treeNode`
`{`
`int data;`
`struct treeNode *left;`
`struct treeNode *right;`
`};`
`treeNode * root;`
	
	Left pointer field Data fieldRight pointer field

Code segment	Memory Drawing

• Create Binary Search Tree
`BinarySearchTree()`
`{`
`root = NULL;`
`}`
	
	root

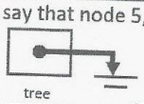
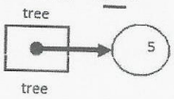
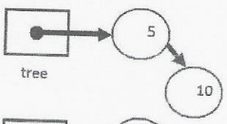
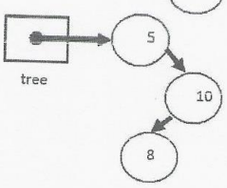
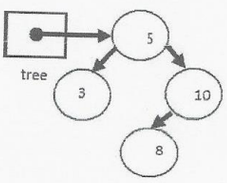
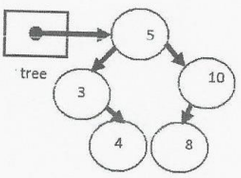
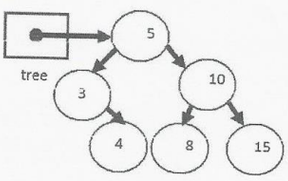
(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

- Based on the code segment above, BST is created when root = NULL. At this time there is no node exist in the BST yet.

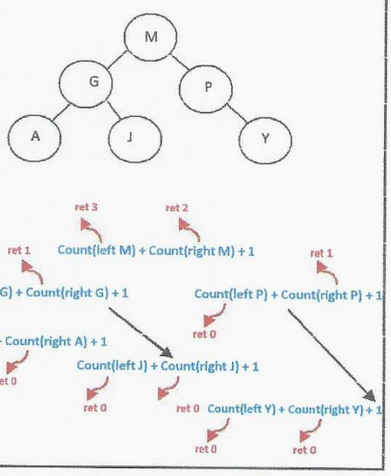
Code segment	Explanation

• Check BST empty?
`int empty()`
`{`
`  if(root == NULL)`
`  return (1);`
`  else`
`  return (0);`
`}`
	• To clarify whether the BST is empty, by checking the value of **root**. If the value is **NULL**, its mean that the BST is empty.

Code segment	Explanation

• Insert new node
`void InsertData(int data)`
`{`
`  Insert(root, data);`
`}`
 	
`void Insert(treeNode*& tree, int data)`
`{`
`  if(tree == NULL)`
`  { // base case`
`  tree = new TreeNode;`
`  tree->right = NULL;`
`  tree->left = NULL;`
`  tree->info = item;`
`  }`
`  else if(item < tree->info)`
`  Insert(tree->left, item);`
`  else`
`  Insert(tree->right, item);`
`}`
	Let say that node 5, 10, 8, 3, 4, 15 used to build BST.
	1. 
	2. 
	3. 
	4. 
	5. 
	6. 
	7. 

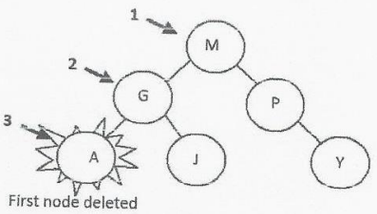
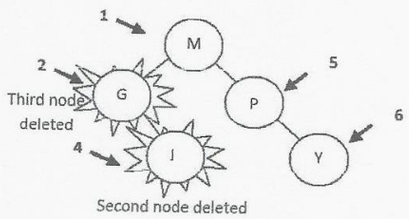
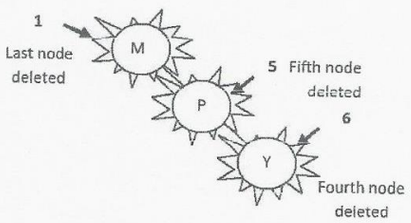
Code segment	Explanation

• Count number of nodes in tree
`int CountNodes(treeNode * tree)`
`{`
`  if (tree == NULL)`
`  return 0;`
`  else`
`  return CountNodes(tree->left)`
`  + CountNodes(tree->right) + 1;`
`}`
	• If **root** is **NULL**, its mean that number of node is **zero (0)** or the BST is empty. If root not NULL, **countNodes()** function will be called recursively to count number of nodes in left subtree + number of nodes in right subtree + 1 (root node).
	 

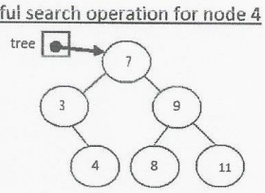
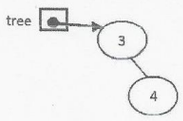
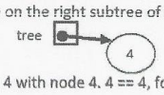
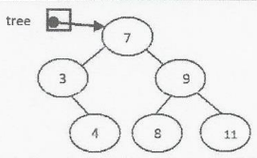
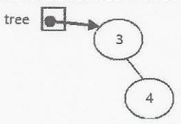
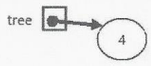
(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Data Structures and Algorithm

Code segment	Explanation

• Remove node from tree
`void Destroy(treeNode*& tree)`
`{`
`  if(tree != NULL)`
`  {`
`  Destroy(tree->left);`
`  Destroy(tree->right);`
`  delete tree;`
`  }`
`}`
	
	First node deleted
	 
	Second node deleted
	 

Code segment	Explanation

• Search nodes in tree
`void retrieve(treeNode * tree, char item, bool& found)`
`{`
`  if (tree == NULL)`
`  found = false;`
`  else if(item < tree->info)`
`  retrieve(tree->left,item,found);`
`  else if(item > tree>info)`
`  retrieve(tree->right,item,found);`
`  else`
`  found = true;`
`}`
	• Search operation used to search node in BST and verify whether the node is existed. Search operation in BST starts from root node and traverse recursively to left subtree and right subtree until the node being found or untill all nodes in BST has been traverse (searching failed).
	• Successful search operation for node 4
	 
	-Compare 4 with node 7. 4 < 7, next search will be done on the left subtree of node 7.
	 
	-Compare 4 with node 3. 4 > 3, next search will be done on the right subtree of node 3.
	 
	-Compare 4 with node 4. 4 == 4, found = True.
	• Unsuccessful search operation for node 5
	 
	-Compare 5 with node 7. 5 < 7, next search will be done on the left subtree of node 7.
	 
	-Compare 5 with node 3. 5 > 3, next search will be done on the right subtree of node 3.
	 
	-Compare 5 with node 4. 5 > 4, next search will be done on the right subtree of node 4.
	 
	-tree = NULL. found = false. There's no 5 in the BST.

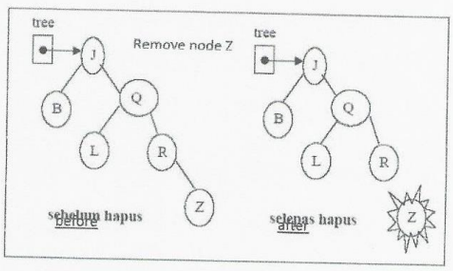
(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Remove item from tree operation

- Removed item from tree operation must be done carefully so that after operation the BST characteristic are still implemented. Before removing item from BST, types of item that want to be deleted has to be clarify. There are 3 condition of item that want to be removed:-
  - Item with no child (Leaf node)
  - Item with 1 child
  - Item with 2 child

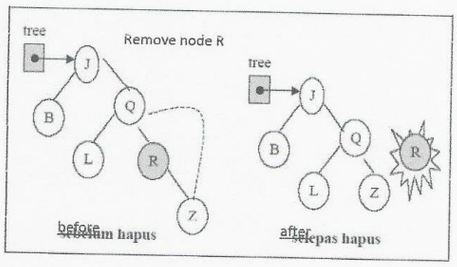
Remove item with no child (Leaf node)

- Leaf node can be removed from BST just like that because there's no relation will be clashed in the BST when removing it.
- Assign NULL value to parent pointer that point to the node that want to be removed.



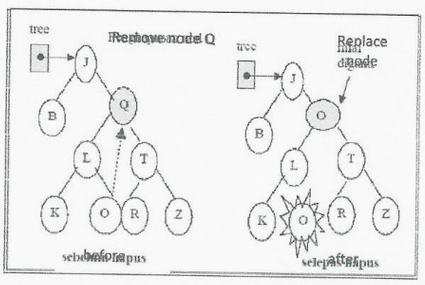
Remove item with 1 child

- If node that want to be remove has 1 child (whether left or right), chain the parent node that want to be removed to the child node that want to be removed. Child node will replace the node that wants to be removed.

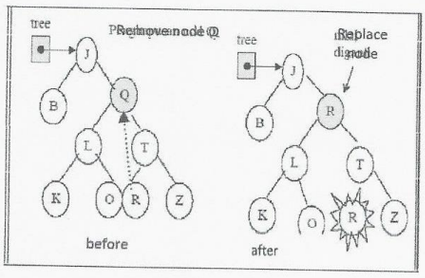


Remove item with 2 child

- Clarify node to replace the removing node. Make sure that the characteristic of BST are still implemented.
- Clarification based on two way:
  1. Find the largest node from left subtree
  2. Find the smallest node from right subtree
- Replace the node that wants to be removed with the node chosen above.



Remove item with 2 child with the largest node from the left subtree



Remove item with 2 child with the smallest node from the right subtree

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Code segment

- Delete item from tree

```
void Delete(treeNode *& tree, int item)
{
  if(item < tree->info)
     Delete(tree->left, item);
  else if(item > tree->info)
     Delete(tree->right, item);
  else
     DeleteNode(tree);
}

void DeleteNode(treeNode *& tree)
{
  int data;
  treeNode * tempPtr;

  // Remove  node  with  1  right
  child tempPtr = tree;
  if(tree->left == NULL)
  { //right child
     tree = tree->right;
     delete tempPtr;
  }

  // Remove node with 1 left child
  else if(tree->right == NULL)
  { // left child
     tree  =  tree->left;
     delete tempPtr;
  }

  // Remove node with 2 child
  else
  {// get predecessor node
     GetPredecessor(tree->left,
     data); tree->info = data;
     Delete(tree->left, data);
  }
}

void GetPredecessor(treeNode * &tree, int & data)
{
  while(tree->right != NULL)
     tree = tree->right; data
     = tree->info;
}
```


(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Activity 3

Based on the sequence of numbers below:

20, 30, 45, 31, 23, 19, 15, 18, 13, 50, 21

a) Draw Binary Search Tree diagram

b) Next, draw new BST to show changes happen for each operation below
(*Node that being removed are no longer exist in the BST)

i. Delete 15
ii. Delete 19
iii. Delete 45
iv. Delete 20

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

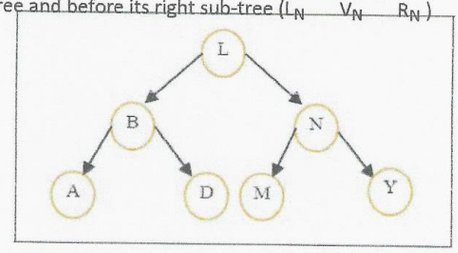
BST traversals

- Once the binary search tree has been created, its elements can be retrieved in-order by recursively traversing the left subtree of the root node, accessing the node itself, then recursively traversing the right subtree of the node, continuing this pattern with each node in the tree as its recursively accessed. The tree may also be traversed in pre-order or post-order traversals.

In-order Traversal

In an inorder traversal a node is visited after its left sub-tree and before its right sub-tree ($L_N \rightarrow V_N \rightarrow R_N$)

```
void inOrder(treeNode tree)
{
    if(tree != NULL)
       inOrder(tree->left);
       cout<< tree->info;
       inOrder(tree->right);
}
```



From the BST diagram, results from an in-order traversal are as below:

A → B → D → L → M → N → Y

Pre-order Traversal

In pre-order traversal a node is visited before its left and right sub-tree ($V_N \rightarrow L_N \rightarrow R_N$)

```
void preOrder(treeNode tree)
{
    if(tree != NULL)
       cout<< tree->info;
       preOrder(tree->left);
       preOrder(tree->right);
}
```

From the BST diagram, results from pre-order traversal are as below:

L → B → A → D → N → M → Y

Post-order Traversal

In post-order traversal a node is visited after its left and right sub-tree ($L_N \rightarrow R_N \rightarrow V_N$)

```
void postOrder(treeNode tree)
{
    if(tree != NULL)
       postOrder(tree->left);
       postOrder(tree->right);
       cout<< tree->info;
}
```

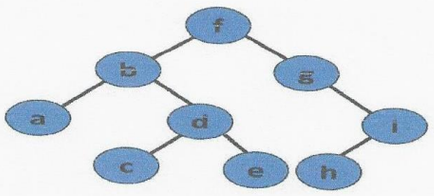
From the BST diagram, results from post-order traversal are as below:

A → D → B → M → Y → N → L

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Activity 4

1. Based on the BST diagram below, write
   a) Pre-Order Traversal
   b) Post-Order Traversal
   c) In-Order Traversal



2. Based on the sequence of words below,

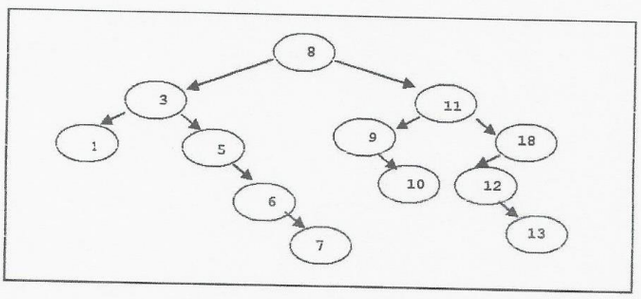
   ALU, ADA, APA, API, ADU, ACI, ABU, ANA

   a) Draw BST diagram
   b) Write pre-order, in-order and post-order traversal

Assessment

Q1. Give definition for Binary Tree.

Q2. Based on the binary tree below, write statements for:



a) Pre-Order Traversal
b) In-Order Traversal
c) Post-Order Traversal

Summary

- A tree is defined as a nonempty finite set of labelled nodes such that there is only one node called the root of the tree, and the remaining nodes are partitioned into subtrees.
- If the tree is either empty or each of its nodes has not more than two subtrees, it is called a *binary tree*.
- Hence each node in a binary tree has either no children, one left child, one right child, or a left child and a right child, each child being the root of a binary tree called a subtree.
- Every node (object) in a binary tree contains information divided into two parts. The first one is proper to the structure of the tree, that is, it contains a key field (the part of information used to order the elements), a parent field, a leftchild field, and a rightchild field. The second part is the object data itself. It can be endogenous (that is, data resides inside the tree) or exogenous (this means that nodes only contains a references to the object's data).
- The root node of the tree has its parent field set to nil. Whenever a node does not have a right child or a left child, then the corresponding field is set to nil.
- A binary search tree is a binary tree with more constraints. If x is a node with key value key[x] and it is not the root of the tree, then the node can have a left child (denoted by left[x]), a right child (right[x]) and a parent (p[x]). If each node of a tree has the following *Binary Search Tree properties*:
  1. for all nodes y in left subtree of x, key[y] < key[x]
  2. for all nodes y in right subtree of x, key[y] > key[x]
- Then this binary tree is called a BINARY SEARCH TREE.

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Belano, Kristel Shaine A.
BSCpE 1-3

96. A parallel circuit is characterized by having the same _____ across every component in the circuit.
A. power B. current
C. voltage D. resistance

97. The total amount of resistance to current flow in a series circuit is equal to the sum of the _____ in that circuit.
A. voltages B. resistances
C. amperages D. wattages

98. _____ are used to provide different voltages between certain points of a circuit.
A. Voltage multipliers B. Resistance adders
C. Voltage dividers D. Voltage adders

99. Series components in a series-parallel circuit may be in series with other _____ components, or with other _____ components.
A. individual, combinations of B. parallel, series
C. series, shunt D. shunt, parallel

100. A circuit in which the resistance is almost zero ohms is referred to by which of the following terms?
A. Broken B. Open
C. Closed D. Short

-- End of Text --

Section 3

INDUCTOR & INDUCTANCE BASIC

Read it till it Hertz!

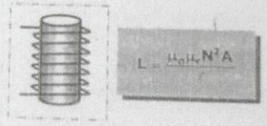
DEFINITION. **Inductance** is the characteristic of an electrical conductor that opposes change in current.

DEFINITION. **Lenz's law:** The induced emf in any circuit is always in a direction to oppose the effect that produced it.

A. INDUCTOR BASIC

1. Inductor

An inductor is a device that stores energy in the magnetic field created around a conductor. The energy is stored in such a way as to oppose any change in current.



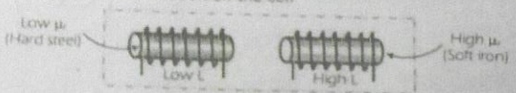
L = Inductance in Henries
μ_0 = permeability of free space
$= 4\pi \times 10^{-7} \frac{H}{m}$
μ_r = relative permeability of core material
N = number of turns
A = area of cross-section of the coil in m^2
ℓ = length of coil in m

Inductance is typified by the behavior of a coil of wire in resisting any change of electric current through the coil.

The symbol L is used for inductance in honor of the physicist Heinrich Lenz. The term inductance was coined by Oliver Heaviside in February 1886. The SI unit of inductance is the henry (symbol: H).

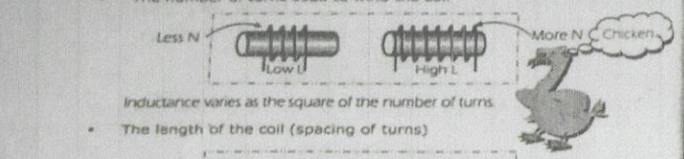
Factors Affecting Coil Inductance

- The type of material used for the core (permeability of the material), and its size and location on the coil



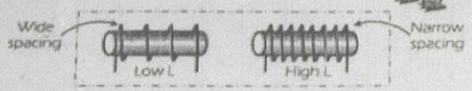
Inductance of a coil increases directly as the permeability of the core material increases.

- The number of turns used to wind the coil



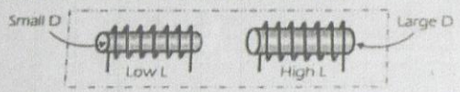
Inductance varies as the square of the number of turns.

- The length of the coil (spacing of turns)



Doubling the length of a coil while keeping the same number of turns halves the value of inductance.

- The diameter of the coil (cross sectional area)



Inductance of a coil increases directly as the cross sectional area of the core increases.

Electrifying Note: An inductor has an inductance of 1 henry if an emf of 1 volt is induced in the inductor when the current through the inductor is changing at the rate of 1 ampere per second.

2. Important qualities of an inductor

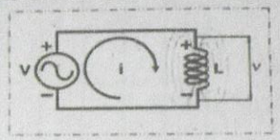
- ★ Current carrying capacity is determined by wire thickness.
- ★ Q, or quality factor, is determined by the uniformity of the windings, as well as the core material and how thoroughly it surrounds the coil.
- ★ Last but not least, the inductance of the coil.

The Q (quality factor) is the ratio of its ability to store energy to the sum total of all energy losses within the component.

B. INDUCTANCE BASIC

1. Inductance

When the current changes in a circuit containing inductance L, the magnetic linkage changes and induces a voltage in the inductance:



$$\frac{d\Psi}{dt} = v = L\frac{di}{dt}$$

- When the current through an inductor is increased, it drops a voltage opposing the direction of electron flow, acting as a **power load**.

  In this condition the inductor is said to be charging, because there is an increasing amount of energy being stored in its magnetic field.

- Conversely, when the current through the inductor is decreased, it drops a voltage aiding the direction of electron flow, acting as a **power source**.

  In this condition the inductor is said to be discharging, because its store of energy is decreasing as it releases energy from its magnetic field to the rest of the circuit.

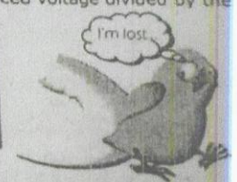
Notes: The induced voltage has a polarity which opposes the rate of change of current.

Alternatively, by integration with respect to time:

$$\Psi = \int v\,dt = LI$$

The inductance L of a circuit is equal to the induced voltage divided by the rate of change of current:

$$L = \frac{v}{\frac{di}{dt}} = \frac{\frac{d\Psi}{dt}}{\frac{di}{dt}} = \frac{d\Psi}{di}$$



(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Alternatively, the inductance L of a circuit is equals to the magnetic linkage Ψ divided by the current I:

$$L = \frac{\Psi}{I}$$

Note that the magnetic linkage Ψ is equal to the product of the number of turns N and the magnetic flux Φ:

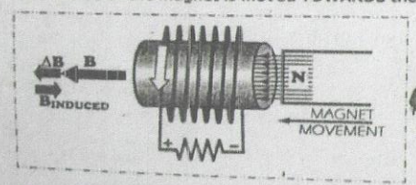
$$\Psi = N\Phi = LI$$

2. **Current flow through an inductor**

- When the current flow is constant, an ideal inductor has no effect on a circuit other than to store energy in the magnetic field in the inductor.
- Whenever the current is increasing, an inductor acts like a **choke** in that it impedes the flow of current by producing a back emf. In this case the inductor acts like a resistance in a circuit in that it causes a voltage drop across the inductor. Some of the energy of the current goes into creating a magnetic field in the inductor.
- Whenever the current is decreasing, an inductor acts like a source of emf (a battery). The energy in the magnetic field is converted to a positive potential creating an additional current flow.

3. **Lenz's Law**
When an emf is generated by a change in magnetic flux according to Faraday's Law, the polarity of the induced emf is such that it produces a current whose magnetic field opposes the change which produces it.

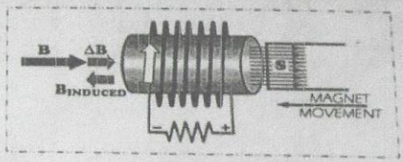
★ **CASE I**
The N-side of the magnet is moved TOWARDS the inductor.



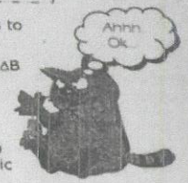
i. The approaching N pole of the magnet causes the magnetic field in the loop to increase.
ii. $B_{INDUCED}$ is always opposite in direction to ΔB as predicted by Lenz's law.

iii. ΔB is in the same direction of B since B is increasing.
iv. The induced magnetic field inside the loop of wire always acts to keep the magnetic flux in the loop constant.

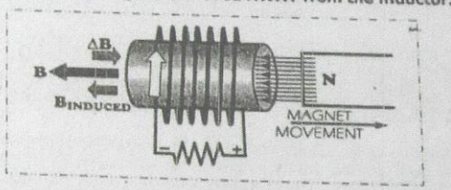
★ **CASE II**
The S-side of the magnet is moved TOWARDS the inductor.



i. The approaching S-side of the magnet tends to increase the B (magnetic field) in the loop.
ii. $B_{INDUCED}$ is always opposite in direction to ΔB as predicted by Lenz's law.
iii. ΔB is in the same direction of B since B is increasing.
iv. The induced magnetic field inside the loop of wire always acts to keep the magnetic flux in the loop constant.

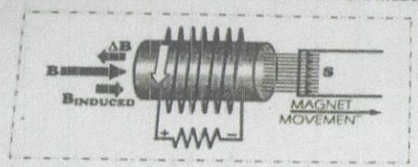


★ **CASE III**
The N-side of the magnet is moved AWAY from the inductor.



i. The receding N-side of the magnet causes the B (magnetic field) in the loop to decrease.
ii. $B_{INDUCED}$ is always opposite in direction to ΔB as predicted by Lenz's law
iii. ΔB is opposite in direction of B since B is decreasing when the magnet is move away from the loop.
iv. The induced magnetic field inside the loop of wire always acts to keep the magnetic flux in the loop constant.

* **CASE IV**
The S-side of the magnet is moved AWAY from the inductor.



i. The receding S-side of the magnet causes the magnetic field (B) in the loop to decrease.

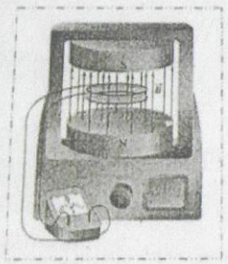
ii. $B_{INDUCED}$ is always opposite in direction to ΔB as predicted by Lenz's law

iii. ΔB is opposite in direction of B since B is decreasing when the magnet is move away from the loop.

iv. The induced magnetic field inside the loop of wire always acts to keep the magnetic flux in the loop constant.

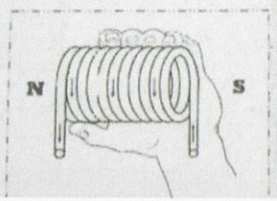
In summary, if the B field is increasing, the induced field acts in opposition to it. If it is decreasing, the induced field acts in the direction of the applied field to try to keep it constant.

When B is constant and shape, location, and orientation of coil does not change, the induced current is zero.

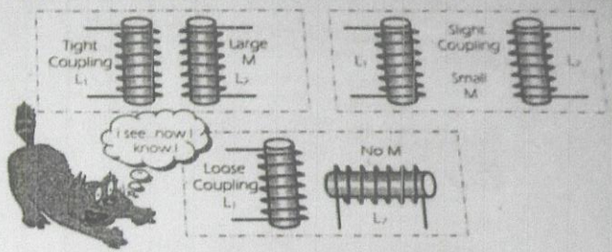


4. Left-hand rule for coils

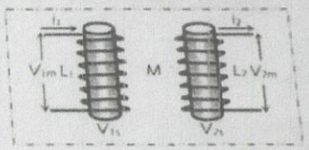
A left-hand rule exists for coils to determine the direction of the magnetic field. The fingers of the left hand are wrapped around the coil in the direction of electron flow. The thumb points to the north pole of the coil.



5. Mutual Inductance (M)
Mutual inductance is the voltage induced in one circuit (the secondary circuit) when the current in another circuit (the primary circuit) changes by a unit amount in unit time.



The mutual inductance M of two coupled inductances L_1 and L_2 is equal to the mutually induced voltage in one inductance divided by the rate of change of current in the other inductance:



$$M = \frac{V_{2m}}{\frac{di_1}{dt}} \qquad M = \frac{V_{1m}}{\frac{di_2}{dt}}$$

If the self induced voltages of the inductances L_1 and L_2 are respectively and V_{2s} for the same rates of change of the current that produced mutually induced voltages V_{1m} and V_{2m}, then:

$$M = L_1\left[\frac{V_{2m}}{V_{1s}}\right] \qquad M = L_2\left[\frac{V_{1m}}{V_{2s}}\right]$$

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

Combining these two equations:

$$M = \sqrt{\frac{V_{1m} \times V_{1m}}{V_{1s} \times V_{2s}}} \times \sqrt{L_1 L_2} = k_M \sqrt{L_1 L_2}$$

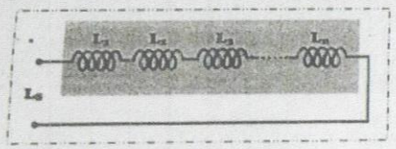
where k_M is the mutual coupling coefficient of the two inductances L_1 and L_2.

Note: The COEFFICIENT OF COUPLING (k_M) between two coils is equal to the ratio of the flux cutting one coil to the flux originated in the other coil.

If the coupling between the two inductances L_1 and L_2 is perfect, then the mutual inductance M is:

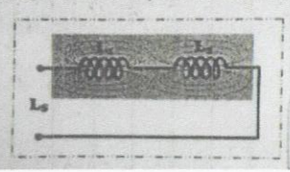
$$M = \sqrt{L_1 L_2}$$

6. **Inductances in Series (Without magnetic coupling)**
When uncoupled inductances L_1, L_2, L_3, ...L_n are connected in series, the total inductance L_S is:



$$L_s = L_1 + L_2 + L_3 + \ldots L_n$$

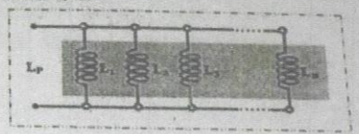
Series Inductors with Magnetic Coupling
When two coupled inductances L_1 and L_2 with mutual inductance M are connected in series, the total inductance L_S is:



$$L_s = L_1 + L_2 \pm 2M$$

The plus or minus sign indicates that th coupling is either additive (series aiding) subtractive (series opposing), depending the connection polarity

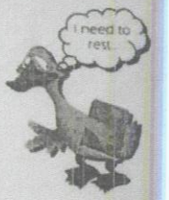
7. **Inductances in Parallel (Without magnetic coupling)**
When uncoupled inductances L_1, L_2, L_3, ...L_n are connected in parallel, the total inductance L_P is:



$$\frac{1}{L_p} = \frac{1}{L_1} + \frac{1}{L_2} + \frac{1}{L_3} + \frac{1}{L_n}$$

8. **Power (P)**
The power P transferred by an inductance L carrying a changing current I with magnetic linkage Ψ is:

In Terms of	Formula
Inductance & Current	$LI\left(\frac{di}{dt}\right)$
Magnetic Linkage & Current	$\Psi\left(\frac{di}{dt}\right)$
Magnetic Linkage & Inductance	$\frac{\Psi\left(\frac{dq}{dt}\right)}{L}$



9. **Magnetic Field (B)**

$$B = \mu \frac{N}{\ell} I$$

where:
B = magnetic field develop in an inductor
μ = permeability
N = # of turns
ℓ = length of inductor

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

19/10/2019

ST. CATHERINE COLLEGE OF VALENZUELA LEARNING MANAGEMENT SYSTEM (SCCV-LMS)

Segarra, Monica M.
Reyes, Don Cedric V.

Dr. Velasco

Methods of Engineering Research

Background

Web-based learning (also known as "E-Learning") is currently one of the major applications of the Internet. "Better technology means better learning". By applying different learning techniques online, such as presentations, assignments, activities, experiments, and quizzes, the students can actively participate anytime and anywhere.

In line with this, the researchers came up with an idea of a system that will help both students and teachers by making learning effortless. The system will provide the teachers to create and deliver content, monitor student participation, and assess student performance. As for the students, they can acquire learning materials, monitor results of their activities, and extend their knowledge with their classmates and teachers.

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Statement of the Problem

1. What are the steps undertaken in the development of St. Catherine College of Valenzuela Learning Management System.

2. What are the effects of Learning Management System to the students in terms of:
   - 2.1 Academic performance
   - 2.2 Motivation to study

3. What is the perception of the students and teachers of St. Catherine College of Valenzuela to the Learning Management System in terms of:
   - 3.1 Functionality
   - 3.2 Usability
   - 3.3 Reliability
   - 3.4 Performance
   - 3.5 Security

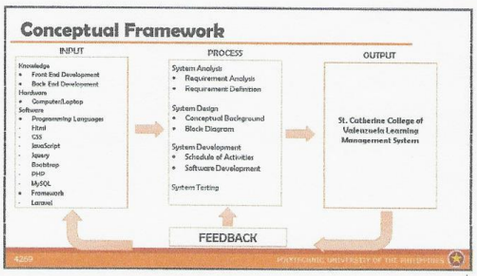
4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Theoretical Framework

- CSS
- HTML
- Javascript
- Bootstrap
- PHP
- MySQL
- Laravel
- Operating System
- Mac OS
- Linux OS

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Conceptual Framework



4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Significance of the Study

1. The study will benefit most the Grade 7-12 students of St. Catherine College of Valenzuela in accessing the lectures, submitting, and answering assignments, online quizzes/exams and other activities online.
2. Teachers/Professors will be able to provide online lectures and other activities to their students.
3. The parents will be able to monitor the grades from quizzes and exams and attendance of their children.
4. The Department of Education will be able to use the Learning Management System to other schools.
5. The school will be able to cope up with the new Learning System.
6. The future researchers will be able to use this study for future reference.

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

19/10/2019

Scope and Limitations

SCOPE

Teacher
- Send and receive messages
- Input record of the activities, seatwork, quizzes and exams. Also the attendance.
- Upload lecture such as pictures, powerpoint, videos, pdf, word and more.
- Input the grades quarterly and final grade

Students
- Send and receive message
- See their records of the activities, seatwork, quizzes and exams.
- Download and upload lecture/assignments
- Answer online activities/quizzes
- Give evaluation to their teachers

Parents/Guardian
- Send and receive messages
- See attendance of the student

Admin
- add/edit /manage records and evaluation.

LIMITATION

Teacher
- Cannot send and receive messages using voice or video call.
- Up to 10 activities every quarter
- Cannot send file using file extensions such as .exe, .msp, .bar, .bat, .dll, .bmp, and etc.

Students
- Not allowed to see his/her classmate's grades
- Given activities on the LMS can only be answered online to avoid cheating.

Parents/Guardians
- Not allowed to see the grades of other students
- Not allowed to see the lectures that are uploaded by the teachers

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Hypothesis

There is an effectiveness of Learning Management System if most of the students and faculties have access to the internet. Using blended learning and eLearning provides additional tools to explain and it catches the attention of the students. LMS is very efficient to use, especially Philippines is part of the typhoon belt. Students in the Philippines always experience suspension of classes, which learning management system will help the students to cope up with activities and others.

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Methods of Research

In order to improve the educational system, the proponents used descriptive research. Descriptive research describes data and characteristics about the population or phenomenon being studied. It aimed at casting light on current issues or problems through a process of data collection that enables them to describe the situation more completely than was possible without employing this method. Most of the colleges and universities in Philippines are using blended learning. The LMS can be effective because the availability of learning was made more efficient by adding collaborative learning tools for students.

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Population and Sample Size

The population of our target respondents is the total population of the Teachers and Students Grade 7 to Grade 12 at St. Catherine College of Valenzuela. The sample size will be from the students who participated in the survey given to them and answered back.

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Description of Respondents

The respondents of the study are the teachers and students from St. Catherine College of Valenzuela. They should be currently teaching and enrolled at the said school.

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Data Gathering Procedure

In this study survey questionnaires will be used. Survey will provide insights and feedback on the project from the respondents as well as to compare the current situation of the existing system currently being implemented. Questionnaires will be used to ask the respondents basic questions that are important for the proponents to use in their study.

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

19/10/2019

Statistical Treatment of Data

The following statistical analysis will be used in interpretation of the data gathered by the proponents:

a. Weighted Mean - The weighted mean determines the weight of each parameter against each other in a parameter.

b. Percentage - In this statistical instrument, the percentage of the respondents is measured.

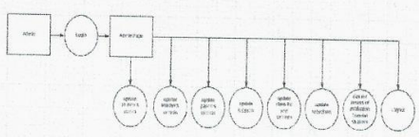
c. Tabular Presentation – This will show the datum computed in tabular form.

Getting the mean through the likert scale between two different section and test the hypothesis between two means.

4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Data Flow Diagram

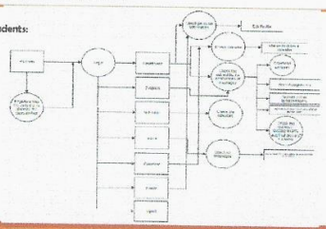
For Admin:



4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Data Flow Diagram

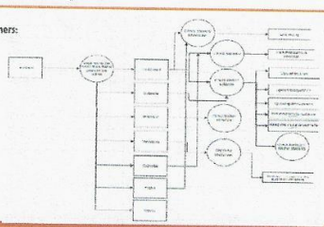
For Students:



4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Data Flow Diagram

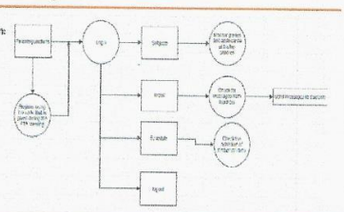
For Teachers:



4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

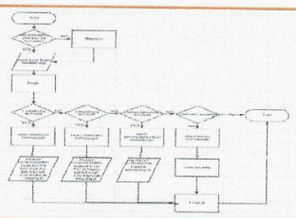
Data Flow Diagram

For Parents/Guardian:



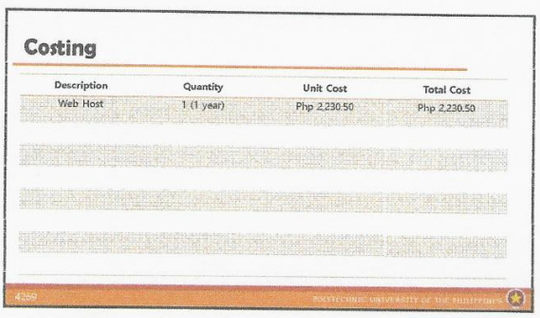
4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Flow Chart



4269 POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

19/10/2019



Costing

Description	Quantity	Unit Cost	Total Cost
Web Host	1 (1 year)	Php 2,230.50	Php 2,230.50

(02) 8713 5968 | dcoe_chair@gmail.com
RM322 CEA BLDG. NDC COMPOUND,
ANONAS COR. PUREZA STREETS, STA. MESA, MANILA

19/10/2019

FAR ULTRA-VIOLET C PHOTO CATALYST MOSQUITO AND FLY EXTERMINATOR LAMP

ALVARADE, CAMILLE JOY C.
BAGTASOS, JESSA A.
BALINGIT, WILLIAM HERBERT C.
OMINGA, JOHN MICHAEL R.

Dr. ARVIN DELA CRUZ

Methods of Engineering Research

Background

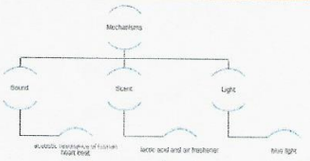
Group Code — POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Statement of the Problem

- What are the requirements needed for implying the Far Ultra Violet C Photo Catalyst Mosquito and Fly Exterminator Lamp?
- What are the stages in creating a device that attract and kills mosquito and fly?
- What is the evaluation of the respondent and clients in using the Far Ultra Violet C Photo Catalyst Mosquito and Fly Exterminator Lamp in terms of: Functionality, Usability, Reliability, Performance Efficiency, and Sustainability.
- What are the problems encountered based on the evaluation of the respondents and clients?
- What are the possible solutions needed to improve Far Ultra Violet C Photo Catalyst Mosquito and Fly Exterminator Lamp?
- What user's manual can be formulated to help in using the Far Ultra Violet C Photo Catalyst Mosquito and Fly Exterminator Lamp?

Group Code — POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

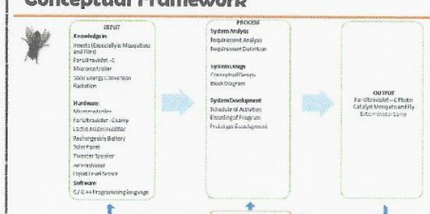
Theoretical Framework



Theoretical framework of how the mechanisms of the Far Ultra Violet C Photo Catalyst Mosquito and Fly Exterminator works.

Group Code — POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Conceptual Framework



Group Code — POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Significance of the Study

Why should we conduct the study?

Group Code — POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

19/10/2019

Scope and Limitations

- The device can attract mosquitoes and flies which are risks to the health of human
- The study concentrates on eliminating these insects without further damage to the environment and waste by-products
- The lamp can contain up to 500 grams of insect remains
- The lamp has a 12v lithium ion battery that can be charged using electricity or solar energy.

Hypothesis

If the study will be conducted using the 3 components that will attract mosquito and fly then large number of these insects will be exterminated and if large numbers of mosquitoes and flies will be exterminated then numbers of people that can have diseases from this insect will be lessen.

Methods of Research

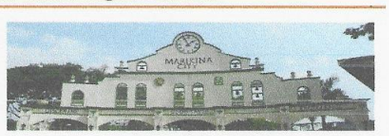
Experimental Research

Population and Sample Size

In this study, the target population for the information gathering are as follows; two entomologists, two radiologists and one producer of lamp. The target population will be the guide of the researchers on consulting experts for their project.

Description of Respondents

Data Gathering Procedure



The researchers will survey in Marikina City and the data will be recorded and evaluated. The data will be analyzed and interpreted according to the specific problems set forth.

19/10/2019

Statistical Treatment of Data

A statistical tool of the study where as the questionnaire consists of questions answerable by statements with Likert Scale. To analyze these type of questions, the following formulas will be used:

For questions answerable by yes or no:

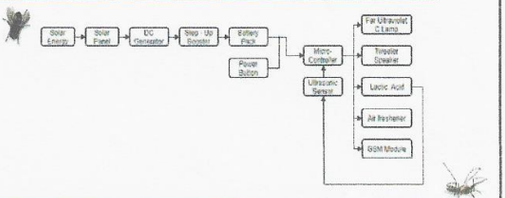
$$(\%)\ \text{Percentage} = (f/n) \times 100$$

$$\%Y = (R\_y/T\_r) \times 100$$

$$\%N = (R\_n/T\_r) \times 100$$

Group Code POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Block Diagram



Group Code POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

Costing

Description	Quantity	Unit Cost	Total Cost
Lactic Acid	1	P250.00	P250.00
Tweeter Speakers	1	P85.00	P85.00
220 nm LED/Bulb	1	P255.00	P255.00
Lithium Battery	4	P88.00	P352.00
Solar Panel	1	P887.19	P887.19
Air Freshener	1	P495.00	P495.00

Group Code POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

THANK YOU!

Group Code POLYTECHNIC UNIVERSITY OF THE PHILIPPINES