Data Structures With C/C++ Lab Manual

Akash Devaraju

[DEPARTMENT OF CS/IS]

2011

Treating Code As An Essay

PROGRAMS SHARE SOME ATTRIBUTES WITH ESSAYS. For essays, the most important question readers ask is, "What is it about?" For programs, the main question is, "What does it do?" In fact, the purpose should be sufficiently clear that neither question ever needs to be uttered. Still, for both essays and computer code, it's always important to look at how each one is written. Even if the idea itself is good, it will be difficult to transmit to the desired audience if it is difficult to understand. The style in which they are written is just as important as their purpose. Both essays and lines of code are meant—before all else—to be read and understood by human beings.

Computers can, of course, deal with complexity without complaint, but this is not the case for human beings. Unreadable code will reduce most people's productivity significantly. On the other hand, easily understandable code will increase it. And we see beauty in such code.

"Beautiful Code" is not an abstract virtue that exists independent of its programmer's efforts. Rather, beautiful code is really meant to help the programmer be happy & productive.

In other words, what is beautiful code?

Brevity: "Succinctness is power", there's a definite cost involved in scanning code with the human eye, and programs should ideally contain no unnecessary information. Eliminate Redundancy …

DRY Principle: Don't Repeat Yourself. If the same code exists in multiple places whatever you are trying to say becomes obscured …

Simplicity: Feel beauty in simple code. If a program is hard to understand, it can't be beautiful. And when programs are obscure rather than comprehensible, the results are bugs, mistakes & confusion.

Flexibility: "Freedom from enforcement of tools". Humans are more valuable than any tools or languages, computers should serve programmers to maximise their productivity & happiness but in reality they often increase the burden instead of lightening it.

Balance is the final element of beautiful code. None of the above elements will by itself ensure a beautiful program. When balanced together & kept in mind from the very beginning, each element will work harmoniously with the others to create a beautiful code. And if you make sure to have fun writing and reading code, you will experience happiness as a programmer ….

Yukihiro ('Matz') Matsumoto

(Creator of the "Ruby" Language)

(Adapted from the book – "The Beautiful Code")

Contents

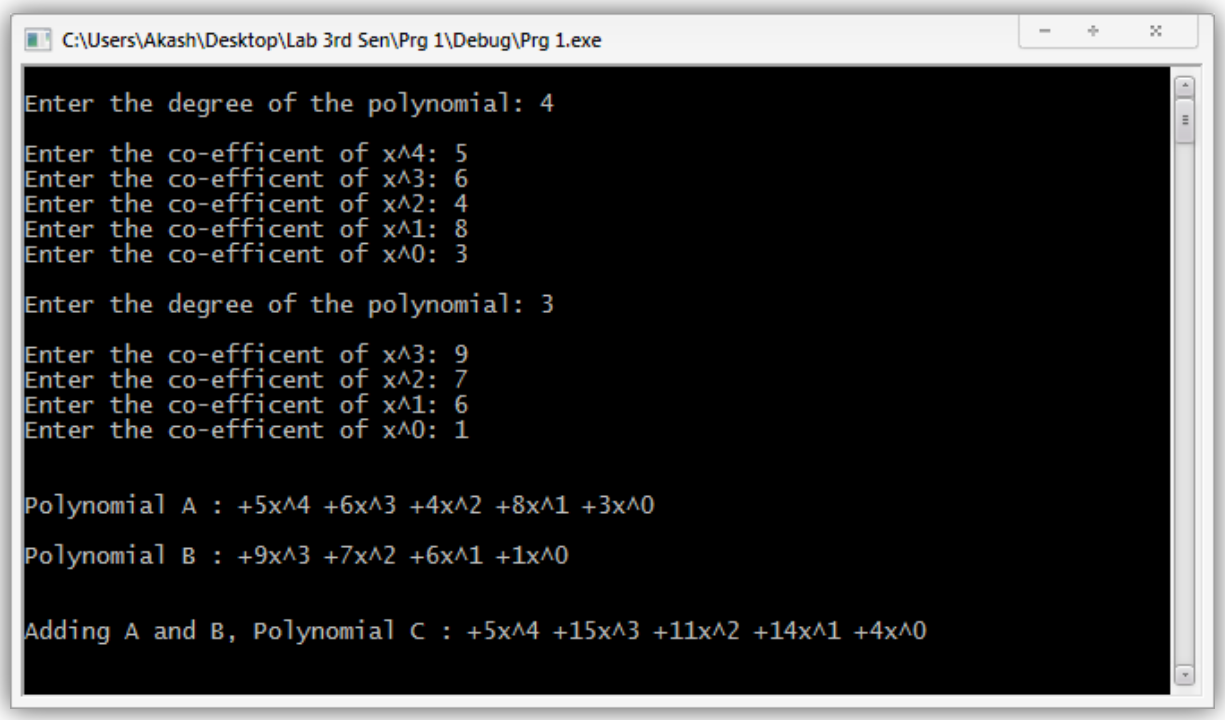
Lab Program 1

```c
/**
Lab Program 1
Circular Polynomial Addition
*/

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

typedef struct Node
{
    int exp;
    int coef;
    struct Node* link;
}Node;

typedef struct
{
    Node* first;
    Node* last;
}Polynomial;

Polynomial Insert(Polynomial P, int exp, int coef);
Polynomial Input(Polynomial P);
void Print(Polynomial P);
Polynomial Add(Polynomial A, Polynomial B, Polynomial C);

void main()
{
    Polynomial a, b, c;
    a.first = b.first = c.first = NULL;

    a = Input(a);
    b = Input(b);

    printf("\n\nPolynomial A : ");
    Print(a);

    printf("\nPolynomial B : ");
    Print(b);

    c = Add(a, b, c);

    printf("\n\nAdding A and B, Polynomial C : ");
    Print(c);

    getch();
}
```

```c
Polynomial Insert(Polynomial P, int exp, int coef)
{
    if(P.first==NULL)
    {
        P.first = P.last =(Node *)malloc(sizeof(Node));
        P.last->link = P.first;
    }
    else
    {
        P.last->link =(Node *)malloc(sizeof(Node));
        P.last = P.last->link;
        P.last->link = P.first;
    }

    P.last->exp = exp;
    P.last->coef = coef;

    return P;
}

Polynomial Input(Polynomial P)
{
    int degree, coef, i;
    printf("\nEnter the degree of the polynomial: ");
    scanf("%d", &degree);

    printf("\n");
    for(i = degree; i >= 0; i--)
    {
        printf("Enter the co-efficent of x^%d: ", i);
        scanf("%d", &coef);
        P = Insert(P, i, coef);
    }

    return P;
}

void Print(Polynomial P)
{
    Node* temp = P.first;
    do
    {
        if(temp->coef == 0)
        {
            temp = temp->link;
            continue;
        }

        printf("%+dx^%d ", temp->coef, temp->exp);
        temp = temp->link;
    }while(temp != P.first);

    printf("\n");
}
```

```
Polynomial Add(Polynomial A, Polynomial B, Polynomial C)
{
    Node* tempA = A.first;
    Node* tempB = B.first;
    do
    {
        if(tempA->exp > tempB->exp)
        {
            C = Insert(C, tempA->exp, tempA->coef);
            tempA = tempA->link;
        }

        else if(tempA->exp == tempB->exp)
        {
            C = Insert(C, tempA->exp, tempA->coef + tempB->coef);
            tempA = tempA->link;
            tempB = tempB->link;
        }

        else
        {
            C = Insert(C, tempB->exp, tempB->coef);
            tempB = tempB->link;
        }

    }while(tempA != A.first || tempB != B.first);

    return C;
}
```

Output

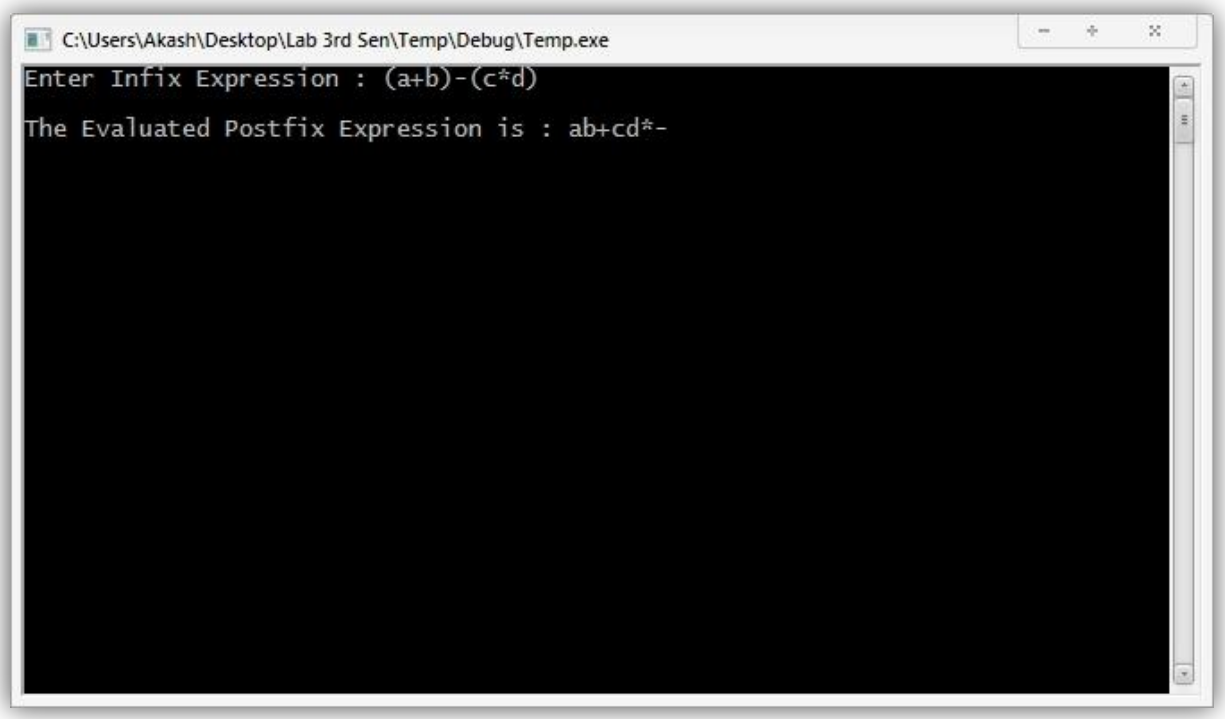

C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 1\Debug\Prg 1.exe
Enter the degree of the polynomial: 4
Enter the co-efficent of x^4: 5
Enter the co-efficent of x^3: 6
Enter the co-efficent of x^2: 4
Enter the co-efficent of x^1: 8
Enter the co-efficent of x^0: 3
Enter the degree of the polynomial: 3
Enter the co-efficent of x^3: 9
Enter the co-efficent of x^2: 7
Enter the co-efficent of x^1: 6
Enter the co-efficent of x^0: 1
Polynomial A : +5x^4 +6x^3 +4x^2 +8x^1 +3x^0
Polynomial B : +9x^3 +7x^2 +6x^1 +1x^0
Adding A and B, Polynomial C : +5x^4 +15x^3 +11x^2 +14x^1 +4x^0

Lab Program 2

```c
/**
Lab Program 2
Infix To Postfix Expression
Author SkyKOG
*/

#include <stdio.h>
#include <conio.h>

int stack[10];
int top=-1;

void push(char);
char pop();
int stpri(char ch);
int inpri(char ch);
void convert(char [],char[]);

void main()
{
    char infix[20],postfix[20];
    push('#');
    printf("Enter Infix Expression : ");
    fflush(stdin);
    gets(infix);
    convert(infix,postfix);
    printf("\nThe Evaluated Postfix Expression is : %s",postfix);
    getch();
}

void push(char ch)
{
    stack[++top]=ch;
}

char pop()
{
    return (stack[top--]);
}

int stpri(char ch)
{
    switch(ch)
    {
    case '#':
    case '(':return 0;
    case '+':
    case '-':return 2;
    case '*':
    case '/':return 4;
    case '$':return 5;
    }
}
```

```
int inpri(char ch)
{
    switch(ch)
    {
    case '+':
    case '-':return 1;
    case '*':
    case '/':return 3;
    case '$':return 6;
    }
}

void convert(char inf[20],char post[20])
{
    int i,j=0;
    char ch;
    for(i=0;inf[i];i++)
    {
        ch=inf[i];
        switch(ch)
        {
        case '(':push(ch);
            break;
        case ')':while(stack[top]!='(')
                    post[j++]=pop();
                pop();
                break;
        case '+':
        case '-':
        case '/':
        case '*':
        case '$':while(stpri(stack[top])>inpri(ch))
                    post[j++]=pop();
            push(ch);
            break;
        default:post[j++]=ch;
        }
    }
    while(stack[top]!='#')
        post[j++]=pop();
    post[j]='\0';
}
```

Output

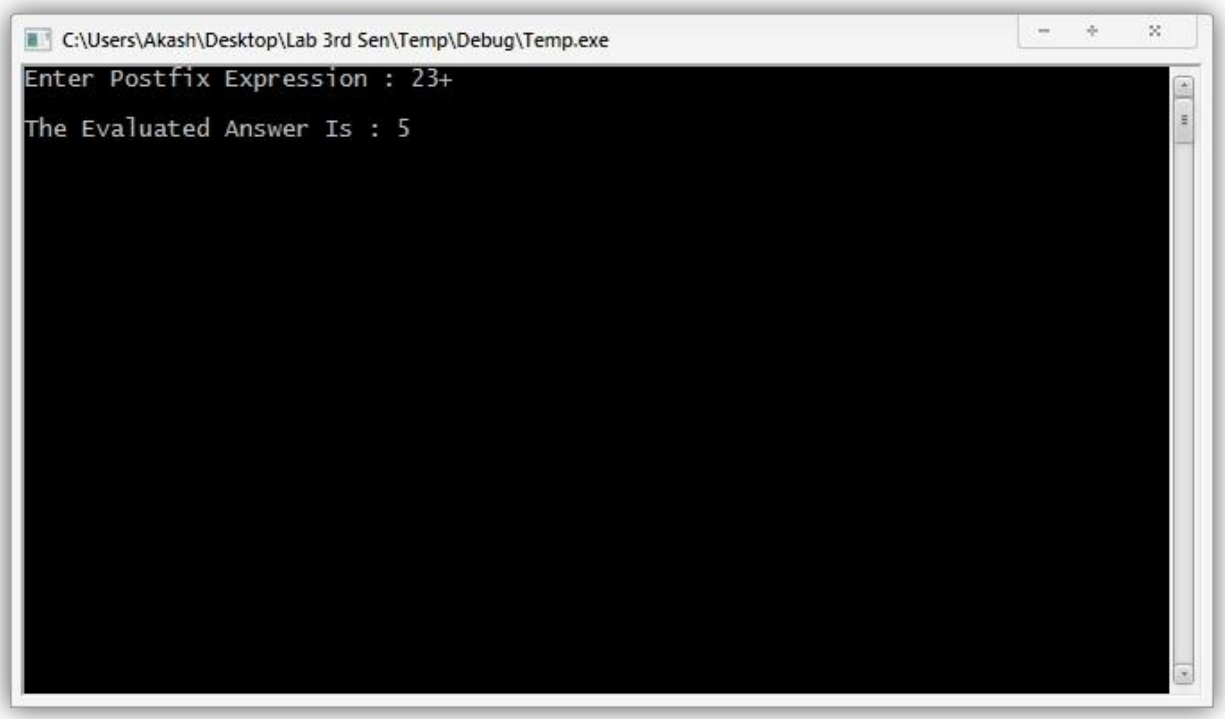

C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
Enter Infix Expression : (a+b)-(c*d)
The Evaluated Postfix Expression is : ab+cd*-

Lab Program 3

```c
/**
Lab Program 3
Postfix Evaluation Using Stack
Author SkyKOG
*/

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <ctype.h>
#include <process.h>

int stack[20];
int top;

void push(int);
int pop();
void eval(char post[20]);

void main()
{
    char post[20];
    top=-1;
    printf("Enter Postfix Expression : ");
    gets(post);
    eval(post);
    printf("\nThe Evaluated Answer Is : %d",pop());
    getch();
}
void push(int ele)
{
    stack[++top]=ele;
}

int pop()
{
    return (stack[top--]);
}

void eval(char post[20])
{
    char ch;
    int i,b,a;
    for(i=0;post[i];i++)
    {
        ch=post[i];
        if(isdigit(ch))
            push(ch-'0');
        else
        {
            b=pop();
            a=pop();
```

```
53                  switch(ch)
54                  {
55                  case '+':push(a+b);
56                      break;
57                  case '-':push(a-b);
58                      break;
59                  case '*':push(a*b);
60                      break;
61                  case '/':if(b==0)
62                           {
63                               printf("Devide By Zero");
64                               getch();
65                               exit(0);
66                           }
67                           push(a/b);
68                  default:printf("Invalid Operation");
69                  }
70              }
71          }
72      }
```

Output


C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
Enter Postfix Expression : 23+
The Evaluated Answer Is : 5

Lab Program 4

```c
/**
Lab Program 4
Implementation of Linear Queue
Author SkyKOG
*/

#include <stdio.h>
#include <conio.h>
#include <windows.h>
#include <process.h>

#define MAX 5

struct queue
{
    int arr[MAX];
    int f,r;
};

void insq(struct queue *,int);
void delq(struct queue *);
void display(struct queue *);

void main()
{
    int ch,ele;
    struct queue q;
    q.f=-1;
    q.r=0;
    while(true)
    {
        system("cls");
        printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n\nEnter Selection : ");
        scanf("%d",&ch);
        switch(ch)
        {
        case 1:printf("\nEnter Element To Insert : ");
            scanf("%d",&ele);
            insq(&q,ele);
            break;
        case 2:delq(&q);
            break;
        case 3:display(&q);
            break;
        case 4:exit(0);
            break;
        default:printf("Invalid Selection");
        }
        getch();
    }
}
```


"A Designer knows he has achieved Perfection not when there's nothing left to add … but when there's nothing left to take away"

```c
void insq(struct queue *q,int ele)
{
    if(q->r==MAX)
    {
        printf("\nQueue Full");
        return;
    }
    q->arr[q->r]=ele;
    q->r++;
    if(q->f==-1)
        q->f=0;
}
void delq(struct queue *q)
{
    if(q->f==-1)
    {
        printf("\nQueue Empty");
        return;
    }
    printf("\nElement Deleted is : %d",q->arr[q->f]);
    q->f++;
    if(q->f==q->r)
    {
        q->f=-1;
        q->r=0;
    }
}
void display(struct queue *q)
{
    int i;
    if(q->f==-1)
    {
        printf("Empty Queue");
        return;
    }
    printf("\nThe Contents of the Queue Are : ");
    for(i=q->f;i<q->r;i++)
    {
        printf("%d ",q->arr[i]);
    }
}
```

Output

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert
2. Delete
3. Display
4. Exit

Enter Selection : 1

Enter Element To Insert : 50
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert
2. Delete
3. Display
4. Exit

Enter Selection : 3

The Contents of the Queue Are : 50 40 20 10 _
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert
2. Delete
3. Display
4. Exit

Enter Selection : 2

Element Deleted is : 10_
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert
2. Delete
3. Display
4. Exit

Enter Selection : 2

Queue Empty
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert
2. Delete
3. Display
4. Exit

Enter Selection : 1

Enter Element To Insert : 90

Queue Full
```

Lab Program 5

```cpp
/**
Lab Program 5
Simple Class Simulation 1
Author SkyKOG
*/

#include <iostream>
#include <Windows.h>
#include <conio.h>
#include <stdlib.h>

using namespace std;

class EMPLOYEE {
    int Employee_Number,Basic_Salary,All_Allowances,IT,Net_Salary;
    char Employee_Name[20];

    public:
        void accept();
        void calc();
        void display();
};

int main()
{
    system("cls");
    EMPLOYEE emp[20];
    int n;
    cout<<"Enter Number Of Employees : ";
    cin>>n;
    for(int i=0 ; i < n; i++)
    {
        cout<<"\nEnter Details for Emp : "<<i+1<<"\n";
        emp[i].accept();
    }

    cout<<"\n\nNumber\tName\tBasic\tIT\tNet\tAll Allowances\n\n";
    for(int i=0 ; i < n ; i++)
    {
        emp[i].display();
    }

    getch();
    return 0;
}
```

```
47  void EMPLOYEE::accept()
48  {
49      cout<<"Number : ";
50      cin>>Employee_Number;
51      cout<<"Name : ";
52      cin>>Employee_Name;
53      cout<<"Basic : ";
54      cin>>Basic_Salary;
55      calc();
56  }
57
58  void EMPLOYEE::calc()
59  {
60      All_Allowances=1.23*Basic_Salary;
61      IT=0.3*(Basic_Salary+All_Allowances);
62      Net_Salary=Basic_Salary+All_Allowances-IT;
63  }
64
65  void EMPLOYEE::display()
66  {
67      cout<<Employee_Number<<"\t"<<Employee_Name<<"\t"<<Basic_Salary;
68      cout<<"\t"<<IT<<"\t"<<Net_Salary<<"\t"<<All_Allowances<<"\n";
69  }
```

Output

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
Enter Number Of Employees : 3

Enter Details for Emp : 1
Number : 23
Name : Brad
Basic : 60000

Enter Details for Emp : 2
Number : 65
Name : Chad
Basic : 90000

Enter Details for Emp : 3
Number : 99
Name : Vlad
Basic : 80000


Number  Name    Basic   IT      Net     All Allowances

23      Brad    60000   40140   93660   73800
65      Chad    90000   60210   140490  110700
99      Vlad    80000   53520   124880  98400
```

Lab Program 6

```cpp
/**
Lab Program 6
Operator Overloading & Copy Constructor - String Class
Author SkyKOG
*/

#include <iostream>
#include <conio.h>
#include <Windows.h>
#include <string>

using namespace std;

class STRING {

    char *str;

    public :
        STRING();
        STRING(char *);
        STRING(const STRING &);
        STRING operator +(STRING &);
        friend ostream & operator <<(ostream &,STRING &);

};

void main()
{
    STRING s1 = "VTU";
    STRING s2 = "BELGAUM";
    STRING s3 = s1 + s2;

    cout<<"s1 = "<<s1<<"\n";
    cout<<"s2 = "<<s2<<"\n";
    cout<<"s3 = "<<s3<<"\n";

    getch();
}

STRING::STRING()
{
    str=NULL;
}

STRING::STRING(char *x)
{
    str=new char[strlen(x)+1];
    strcpy(str,x);
}
```

```
STRING::STRING(const STRING &s3)
{
    str=new char[strlen(s3.str)+1];
    strcpy(str,s3.str);
}

STRING STRING::operator +(STRING &s2)
{
    STRING temp;

    temp.str=new char[strlen(str)+strlen(s2.str)+1];

    strcpy(temp.str,str);
    strcat(temp.str,s2.str);

    return temp;
}

ostream & operator <<(ostream &out,STRING &s)
{
    out<<s.str<<"\n\n";
    return out;
}
```

Output


C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
s1 = VTU
s2 = BELGAUM
s3 = VTUBELGAUM

Lab Program 7

```cpp
/**
Lab Program 7
Operator Overloading - Stack Class
Author SkyKOG
*/

#include <iostream>
#include <conio.h>
#include <process.h>
#include <windows.h>

#define MAX 5

using namespace std;

class STACK {

    int top;
    int arr[MAX];

    public:
        STACK();
        STACK operator + (int);
        STACK operator -- ();
        friend void operator <<(ostream &,STACK &);

};

void main()
{
    STACK s1;
    int ch,ele;

    while(true)
    {
        system("cls");
        cout<<"1. Push\n2. Pop\n3. Display\n4. Exit\n\nEnter Your Choice : ";
        cin>>ch;

        switch(ch)
        {
        case 1:cout<<"\nEnter The Element to Push : ";
            cin>>ele;
            s1 = s1 + ele;
            cout<<"\n"<<s1;
            break;
        case 2:s1 = --s1;
            cout<<"\n"<<s1;
            break;
        case 3:cout<<"\n"<<s1;
            break;
        case 4:exit(0);
            break;
        default:cout<<"\nInvalid Selection\n";
        }
        getch();
    }
}
```

```
STACK::STACK()
{
    top=-1;
}

STACK STACK::operator +(int ele)
{
    if(top==MAX-1)
    {
        cout<<"\nSTACK Overflow";
        return *this;
    }

    arr[++top]=ele;
    return *this;
}


STACK STACK::operator --()
{
    if(top==-1)
    {
        cout<<"\nSTACK Underflow\n";
        return *this;
    }

    cout<<"\nThe Popped Element is : "<<arr[top--]<<"\n";
    return *this;
}


void operator <<(ostream &out,STACK &s)
{
    if(s.top==-1)
    {
        out<<"\nSTACK Empty";
        return;
    }

    cout<<"\nThe Current Contents of STACK Are : ";
    for(int i = s.top ; i >= 0 ; i--)
    {
        out<<s.arr[i]<<" ";
    }
}
```

Output

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Push
2. Pop
3. Display
4. Exit

Enter Your Choice : 1

Enter The Element to Push : 23


The Current Contents of STACK Are : 23
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Push
2. Pop
3. Display
4. Exit

Enter Your Choice : 1

Enter The Element to Push : 77


The Current Contents of STACK Are : 77 89 56 23
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Push
2. Pop
3. Display
4. Exit

Enter Your Choice : 1

Enter The Element to Push : 85

STACK Overflow

The Current Contents of STACK Are : 99 77 89 56 23 _
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Push
2. Pop
3. Display
4. Exit

Enter Your Choice : 2

The Popped Element is : 77


The Current Contents of STACK Are : 89 56 23
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Push
2. Pop
3. Display
4. Exit

Enter Your Choice : 2

The Popped Element is : 23


STACK Empty
```

Lab Program 8

```cpp
/**
Lab Program 8
Linked List
Author SkyKOG
*/

#include <iostream>
#include <Windows.h>
#include <conio.h>

using namespace std;

class LIST {
    struct node
    {
        int item;
        node *link;
    }*p;

    public:
        LIST();
        void insert(int);
        int del(int &);
        void display();
        ~LIST();
};

int main()
{
    LIST obj;
    int ch,ele,status=0;
    while(true)
    {
        system("cls");
        cout<<"1. Insert At Front\n2. Delete From Front\n3. Display\n4. Exit\n\nEnter Choice : ";
        cin>>ch;
        switch(ch)
        {
        case 1:cout<<"\nEnter Element to Insert : ";
            cin>>ele;
            obj.insert(ele);
            break;
        case 2:ele=obj.del(status);
            if(status==0)
            {
                break;
            }
            cout<<"\nElement Deleted is : "<<ele;
            break;
        case 3:obj.display();
            break;
        case 4:exit(0);
            break;
        default:cout<<"Invalid Selection";
            break;
        }getch();
    }
    return 0;
}
```


"Programs must be written for people to understand …. And only incidentally for machines to execute "

```
LIST::LIST()
{
    p=NULL;
}

void LIST::insert(int item)
{
    node *q;
    q=new node;
    q->item=item;
    q->link=p;
    p=q;
}

int LIST::del(int &status)
{
    int del;
    node *q;
    if(p==NULL)
    {
        status=0;
        cout<<"List Is Empty";
        return NULL;
    }

    q=p;
    status=1;
    del=p->item;
    p=p->link;
    delete q;
    return del;
}

void LIST::display()
{
    node *q;
    if(p==NULL)
        {
            cout<<"\nEmpty List";
            return;
        }
    cout<<"\n\nCurrent List Status : ";
    for(q=p ; q!=NULL ; q=q->link)
        cout<<q->item<<" ";
}

LIST::~LIST()
{
    node *q;
    if(p==NULL)
        return;
    while(p!=NULL)
    {
        q=p->link;
        delete p;
        p=q;
    }
}
```

Output

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert At Front
2. Delete From Front
3. Display
4. Exit

Enter Choice : 3


Current List Status : 99 56 89 26
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert At Front
2. Delete From Front
3. Display
4. Exit

Enter Choice : 1

Enter Element to Insert : 100
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert At Front
2. Delete From Front
3. Display
4. Exit

Enter Choice : 3


Current List Status : 100 99 56 89 26
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert At Front
2. Delete From Front
3. Display
4. Exit

Enter Choice : 2

Element Deleted is : 100
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert At Front
2. Delete From Front
3. Display
4. Exit

Enter Choice : 3


Current List Status : 99 56 89 26 _
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Insert At Front
2. Delete From Front
3. Display
4. Exit

Enter Choice : 2

List Is Empty
```

Lab Program 9

```c
/**
Lab Program 9
Sparse Martix Search
Author Strider24/SkyKOG
*/

#include <stdio.h>
#include <conio.h>

typedef struct
{
    int row;
    int column;
    int value;
}Element;

void main()
{
    Element E[10];
    int count, key, i;


    printf("Enter the Number of Non Zero Entities In The Sparse Matrix: ");
    scanf("%d", &count);
    printf("\n");

    for(i = 0; i < count; i++)
    {
        printf("\nEnter Data For Element %d in <row,column,value> Format : ",i+1);
        scanf("%d", &E[i].row);
        scanf("%d", &E[i].column);
        scanf("%d", &E[i].value);
    }

    printf("\n\nEnter the element to search for : ");
    scanf("%d", &key);

    for(i = 0; i < count; i++)
        if(E[i].value == key)
        {
            printf("\nThe element %d Is Found At Row %d And Column %d", key, E[i].row, E[i].column);
            break;
        }

    if(i == count)
        printf("\nElement Not Found");

    getch();
}
```


"In Software the most beautiful code, the most beautiful functions & the most beautiful programs are sometimes not there at all"

Output

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 9\Debug\Prg 9.exe
Enter the Number of Non Zero Entities In The Sparse Matrix: 4

Enter Data For Element 1 in <row,column,value> Format : 6 9 26
Enter Data For Element 2 in <row,column,value> Format : 4 5 96
Enter Data For Element 3 in <row,column,value> Format : 1 2 36
Enter Data For Element 4 in <row,column,value> Format : 698 596 1

Enter the element to search for : 96
The element 96 Is Found At Row 4 And Column 5
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 9\Debug\Prg 9.exe
Enter the Number of Non Zero Entities In The Sparse Matrix: 3

Enter Data For Element 1 in <row,column,value> Format : 6 9 85
Enter Data For Element 2 in <row,column,value> Format : 2 3 65
Enter Data For Element 3 in <row,column,value> Format : 4 5 1

Enter the element to search for : 99
Element Not Found
```

Lab Program 10

```c
/**
Lab Program 10
Heap Operations
Author SkyKOG
*/

#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <Windows.h>

void heaper(int,int*);
void insert(int,int*,int*);
void display(int *,int);

void main()
{
    int arr[20]={1000},ele,n=0,ch;

    while(true)
    {
        system("cls");
        printf("1. Heap Insert\n2. Display Array\n3. Exit\n\nEnter Choice : ");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:printf("\nEnter Element to Insert : ");
            scanf("%d",&ele);
            insert(ele,arr,&n);
            display(arr,n);
            break;
            case 2:display(arr,n);
                break;
            case 3:exit(0);
                break;
            default:printf("Invalid Selection");
                break;
        }
        getch();
    }
}
```

```c
void display(int *arr,int n)
{
    int i;
    printf("\n\nThe Contents of The Heap are as follows : ")
    for(i=1;i<=n;i++)
    {
        printf("%d ",arr[i]);
    }
}

void insert(int ele,int *arr,int *n)
{
    (*n)++;
    arr[*n]=ele;
    heaper(*n,arr);
}

void heaper(int i,int *arr)
{
    int val;
    val=arr[i];
    while(arr[i/2]<=val)
    {
        arr[i]=arr[i/2];
        i=i/2;
    }
    arr[i]=val;
}
```

Output

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Heap Insert
2. Display Array
3. Exit

Enter Choice : 1

Enter Element to Insert : 8


The Contents of The Heap are as follows : 8
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Heap Insert
2. Display Array
3. Exit

Enter Choice : 1

Enter Element to Insert : 17


The Contents of The Heap are as follows : 17 8
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Heap Insert
2. Display Array
3. Exit

Enter Choice : 1

Enter Element to Insert : 63


The Contents of The Heap are as follows : 63 8 17
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Heap Insert
2. Display Array
3. Exit

Enter Choice : 1

Enter Element to Insert : 32


The Contents of The Heap are as follows : 63 32 17 8 _
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Heap Insert
2. Display Array
3. Exit

Enter Choice : 1

Enter Element to Insert : 56


The Contents of The Heap are as follows : 63 56 17 8 32
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
1. Heap Insert
2. Display Array
3. Exit

Enter Choice : 2


The Contents of The Heap are as follows : 63 56 17 8 32
```

Lab Program 11

```c
/**
Lab Program 11
Double Linked List Operations
Author SkyKOG
*/

#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <windows.h>

struct node
{
    int item;
    struct node *l;
    struct node *r;
};

struct node *getnode()
{
    struct node *temp;
    temp=(struct node *)malloc(sizeof(struct node));
    return temp;
}

struct node *addatbeg(struct node *,int);
struct node *del(struct node *);
struct node *addleft(struct node *,int,int);
void display(struct node*);

void main()
{
    struct node *p;
    p=NULL;
    int ele,ch,key=0;
    while(true)
    {
        system("cls");
        printf("1. Insert At Front\n2. Insert to Left\n3. Delete Node\n4. Display\n5. Exit\n\nEnter Choice : ");
        scanf("%d",&ch);
        switch(ch)
        {
        case 1:printf("\nEnter Element To Insert : ");
            scanf("%d",&ele);
            p=addatbeg(p,ele);
            break;
        case 2:printf("\nEnter Element To Insert : ");
            scanf("%d",&ele);
            printf("\nEnter Key To Search On : ");
            scanf("%d",&key);
            p=addleft(p,ele,key);
            break;
        case 3:p=del(p);
            break;
        case 4:display(p);
            break;
        case 5:exit(0);
            break;
        default:printf("\nInvalid Selection");
```

```c
        }
        getch();
    }
}

void display(struct node *q)
{
    struct node *r;
    r=q;
    if(r==NULL)
    {
        printf("\nList Is Empty");
        return;
    }
    printf("\nThe Contents of the List Are : ");
    while(r!=NULL)
    {
        printf("%d ",r->item);
        r=r->r;
    }
    printf("\n");
}

struct node *addatbeg(struct node *q,int ele)
{
    struct node *temp,*r;
    temp=getnode();
    temp->item=ele;
    temp->l=NULL;
    temp->r=NULL;
    if(q==NULL)
    {
        q=temp;
        return q;
    }
    temp->r=q;
    q->l=temp;
    q=temp;
    return q;
}
```

```c
struct node * addleft(struct node *q,int ele,int key)
{
    struct node *r,*temp;
    int flag=0;
    temp=getnode();
    r=q;
    if(r==NULL)
    {
        printf("\nEmpty List");
        return q;
    }
    if(q->item==key)
    {
        q=addatbeg(q,ele);
        return q;
    }
    while(r->r!=NULL)
    {
        if(r->item==key)
        {
            temp->item=ele;
            temp->r=r;
            temp->l=r->l;
            r->l->r=temp;
            r->l=temp;
            flag=1;
        }
        r=r->r;
    }
    if((r->r==NULL)&&(r->item==key))
    {
        temp->item=ele;
        temp->r=r;
        temp->l=r->l;
        r->l->r=temp;
        r->l=temp;
        return q;
    }

    if(!flag)
        printf("\nKey Not Found In List");
    return q;
}
```

```c
struct node * del(struct node *q)
{
    int ele,del,status=0;
    printf("\nEnter Element To Delete : ");
    scanf("%d",&ele);
    struct node *temp;
    temp=q;
    if(temp==NULL)
    {
        status=0;
        printf("\nList Is Empty");
        return NULL;
    }

    if(temp->item==ele)
    {
        status=1;
        q=q->r;
        del=temp->item;
        printf("\nDeleted item is %d",del);
        free(temp);
        return q;
    }
    while(temp->r!=NULL)
    {
        if(temp->item==ele)
        {
            status=1;
            del=temp->item;
            printf("\nDeleted item is %d",del);
            temp->l->r=temp->r;
            temp->r->l=temp->l;
            free(temp);
            return q;
        }
        temp=temp->r;
    }
    if((temp->r==NULL)&&(temp->item==ele))
    {
        temp->l->r=NULL;
        status=1;
        del=temp->item;
        printf("\nDeleted item is %d",del);
        free(temp);
        return q;
    }
    if(!status)
        printf("\nElement Not in List");
    return q;
}
```


"Programs must be written for people to read ….and only incidentally for machines to understand"

Output

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 4

The Contents of the List Are : 80 90 56 10
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 1

Enter Element To Insert : 100
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 4

The Contents of the List Are : 100 80 90 56 10
```


"In Software the most beautiful code, the most beautiful functions & the most beautiful programs are sometimes not there at all"

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 3

Enter Element To Delete : 90

Deleted item is 90
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 4

The Contents of the List Are : 100 80 56 10
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 3

Enter Element To Delete : 99

Element Not in List
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 4

The Contents of the List Are : 80 56 10
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 2

Enter Element To Insert : 70

Enter Key To Search On : 80
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 4

The Contents of the List Are : 70 80 56 10
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 2

Enter Element To Insert : 50

Enter Key To Search On : 56
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 2

Enter Element To Insert : 9

Enter Key To Search On : 10
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 4

The Contents of the List Are : 70 80 50 56 9 10
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 11\Debug\12. Double Link List.exe
1. Insert At Front
2. Insert to Left
3. Delete Node
4. Display
5. Exit

Enter Choice : 2

Enter Element To Insert : 100

Enter Key To Search On : 999

Key Not Found In List
```

Lab Program 12

```
/**
Lab Program 7
Operator Overloading - date Class
Author SkyKOG
*/

#include <iostream>
#include <conio.h>
#include <windows.h>
#include <process.h>

using namespace std;

class date {

    int dd,mm,yy;

    public :
        void getdata();
        int operator -(date);
        date operator +(int);
        friend ostream & operator <<(ostream &,date &);
};

int main()
{
    int no_of_days,ch;
    date d1,d2;
    while(true)
    {
        system("cls");
        cout<<"1. Diffrence Between 2 dates : \n2. Get New date : \n3. Exit\n\nEnter Selection : ";
        cin>>ch;
        switch(ch)
        {
            case 1:cout<<"\n\nEnter date d1 (dd/mm/yy) : ";
                   d1.getdata();
                   cout<<"Enter date d2 (dd/mm/yy) : ";
                   d2.getdata();
                   no_of_days=d1-d2;
                   if(no_of_days<0)
                   {
                       cout<<"\nInvalid dates";
                   }
                   else
                       cout<<"\nThe Number of Days in Between Are : "<<no_of_days;
                   break;
            case 2:cout<<"\n\nEnter date d1 (dd/mm/yy) : ";
                   d1.getdata();
                   cout<<"Enter Days To Add : ";
                   cin>>no_of_days;
                   d2=d1+no_of_days;
                   cout<<"\nThe Required date is : ";
                   cout<<d2;
                   break;
            case 3:exit(0);
                   break;
            default:cout<<"\n\nInvalid Selection";
                   break;
        }
        getch();
    }
    return 0;
}
```

```
void date::getdata()
{
    cin>>dd>>mm>>yy;

    if(dd>31||mm>12)
    {
        cout<<"\nInvalid date";
        getdata();
    }
    if(((mm==4)||(mm==6)||(mm==9)||(mm==11))&&(dd>30))
    {
        cout<<"\nInvalid date";
        getdata();
    }
    if((mm==2))
    {
        if((yy%4!=0)&&(dd>28))
        {
            cout<<"\nInvalid date";
            getdata();
        }
        else if((dd>29))
        {
            cout<<"\nInvalid date";
            getdata();
        }
    }
}

int date::operator -(date d2)
{
    int i,n1,n2,ny,d;
    n1=n2=ny=0;

    for(i=1;i<=mm;i++)
    {
        if((i==4)||(i==6)||(i==9)||(i==11))
            n1+=30;
        else if(i==2)
        {
            if(yy%4==0)
                n1+=29;
            else
                n1+=28;
        }
        else
            n1+=31;
    }
    n1+=dd;

    for(i=1;i<=d2.mm;i++)
    {
         if((i==4)||(i==6)||(i==9)||(i==11))
            n2+=30;
        else if(i==2)
        {
            if(d2.yy%4==0)
                n2+=29;
            else
                n2+=28;
        }
        else
            n2+=31;
    }
    n2+=d2.dd;
    ny=(yy-d2.yy)*365;
    d=n1-n2+ny;
    return d;
```

```
}

date date::operator +(int n)
{
    while(n>365)
    {
        yy++;
        if(yy%4==0)
            n-=366;
        n-=365;
    }
    while(n>30)
    {
        if((mm==4)||(mm==6)||(mm==9)||(mm==11))
            n-=30;
        else if(mm==2)
        {
            if(yy%4==0)
                n-=29;
            else
                n-=28;
        }
        else
            n-=31;
        mm++;
        if(mm>12)
        {
            yy++;
            mm=1;
        }
    }
    dd+=n;
    if(dd>31)
    {
        if((mm==4)||(mm==9)||(mm==6)||(mm==11))
            dd-=30;
        else if(mm==2)
        {
            if(yy%4==0)
                dd-=29;
            else
                dd-=28;
        }
        else
            dd-=31;
        mm++;
    }
    return *this;
}

ostream & operator<<(ostream &out,date &d3)
{
    out<<d3.dd<<" "<<d3.mm<<" "<<d3.yy;
    return out;
}
```

Output

```
C:\Users\Akash\Documents\Visual Studio 2010\Projects\LAB Oops\Prg 7\Debug\Prg 7.exe
1. Diffrence Between 2 dates :
2. Get New date :
3. Exit

Enter Selection : 1


Enter date d1 (dd/mm/yy) : 23 2 1991
Enter date d2 (dd/mm/yy) : 20 2 1991

The Number of Days in Between Are : 3
```

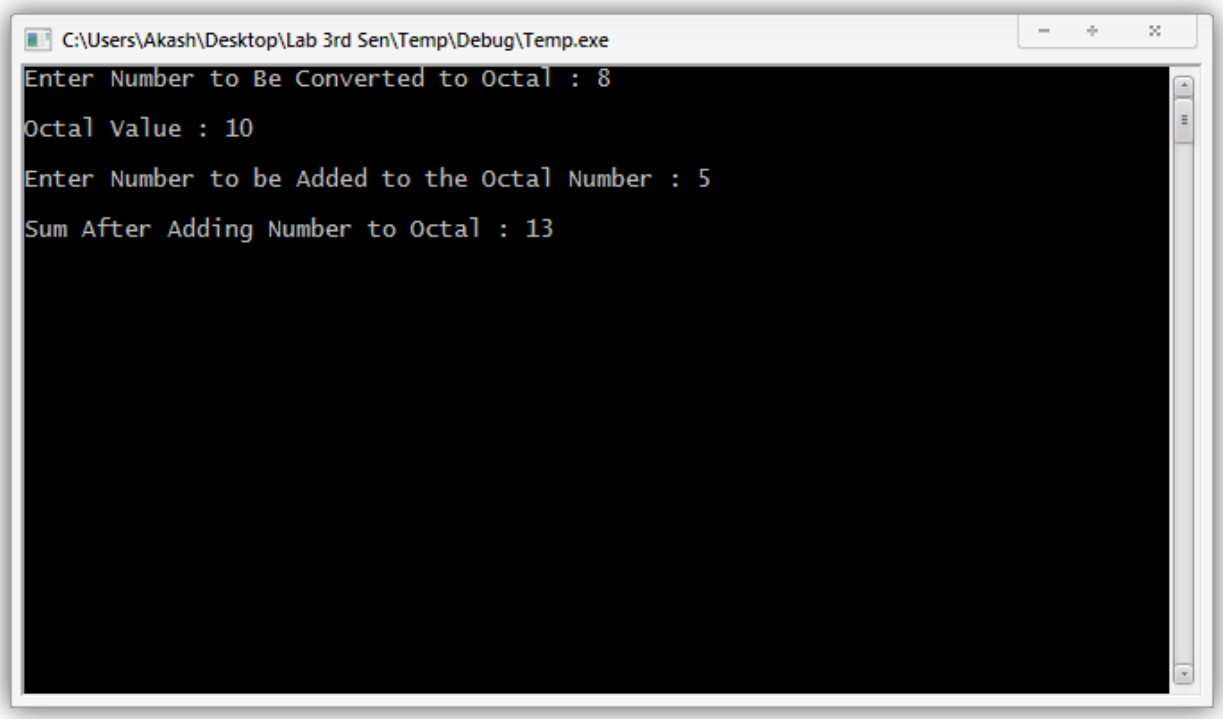
```
C:\Users\Akash\Documents\Visual Studio 2010\Projects\LAB Oops\Prg 7\Debug\Prg 7.exe
1. Diffrence Between 2 dates :
2. Get New date :
3. Exit

Enter Selection : 2


Enter date d1 (dd/mm/yy) : 22 1 1999
Enter Days To Add : 3

The Required date is : 25 1 1999
```

Lab Program 13

```cpp
/**
Lab Program 13
Operator Overloading & Constructor - Octal Class
Author SkyKOG
*/

#include <iostream>
#include <Windows.h>
#include <conio.h>

using namespace std;

class OCTAL {

    int num,y;

    public :
        OCTAL(int);
        int convert();
        int operator +(int);
        friend ostream & operator <<(ostream &,OCTAL &);

};

void main()
{
    int x,k;
    cout<<"Enter Number to Be Converted to Octal : ";
    cin>>x;

    OCTAL h = x;

    cout<<"\nOctal Value : "<<h;
    cout<<"\n\nEnter Number to be Added to the Octal Number : ";
    cin>>k;

    int y = h + k;

    cout<<"\nSum After Adding Number to Octal : "<<y;
    getch();

}

OCTAL::OCTAL(int x)
{
    num=x;
    y=convert();
}
```

```
50 int OCTAL::convert()
51 {
52     int sum=0,rem,product=1,temp=num;
53     while(temp>0)
54     {
55         rem=temp%8;
56         sum+=(product*rem);
57         product*=10;
58         temp/=8;
59     }
60     return sum;
61 }
62
63 int OCTAL::operator +(int k)
64 {
65     return num+k;
66 }
67
68 ostream & operator <<(ostream &out,OCTAL &obj)
69 {
70     out<<obj.y;
71     return out;
72 }
```

Output


C:\Users\Akash\Desktop\Lab 3rd Sen\Temp\Debug\Temp.exe
Enter Number to Be Converted to Octal : 8
Octal Value : 10
Enter Number to be Added to the Octal Number : 5
Sum After Adding Number to Octal : 13

Lab Program 14

```
/**
Lab Program 14
Binary Tree Operations
Author SkyKOG
*/

#include <iostream>
#include <Windows.h>
#include <conio.h>
#include <process.h>

using namespace std;

class BIN_TREE {

    struct node
    {
        node *l;
        node *r;
        int data;
    }*p;

    public:
        BIN_TREE();
        void insert(int);
        void insertmain(node **,int);
        void display(int);
        void inorder(node *);
        void preorder(node *);
        void postorder(node *);

};

BIN_TREE::BIN_TREE()
{
    p=NULL;
}

void main()
{
    BIN_TREE obj;
    int ch,ele;
    while(true)
    {
        system("cls");
        cout<<"1. Insert\n2. Preorder\n3. Inorder\n4. Postorder\n5. Exit\n\nEnter Selection : ";
        cin>>ch;
        switch(ch)
        {
        case 1:cout<<"\nEnter Element To Insert : ";
            cin>>ele;
            obj.insert(ele);
            break;
        case 2:obj.display(2);
            break;
```


Life is Generous to Those who peruse it generously ....

-Martin King

```
56          case 3:obj.display(3);
57              break;
58          case 4:obj.display(4);
59              break;
60          case 5:exit(0);
61              break;
62          default:cout<<"Invalid Selection";
63          }
64          getch();
65      }
66  }
67
68  void BIN_TREE::insert(int ele)
69  {
70      insertmain(&p,ele);
71  }
72
73  void BIN_TREE::insertmain(struct node **q,int ele)
74  {
75      if((*q)==NULL)
76      {
77          (*q)=new node;
78          (*q)->l=NULL;
79          (*q)->r=NULL;
80          (*q)->data=ele;
81          return;
82      }
83
84      else
85      {
86          if(ele==(*q)->data)
87          {
88              cout<<"Duplication Not Allowed";
89              getch();
90              return;
91          }
92          else
93          {
94              if(ele<((*q)->data))
95                  insertmain(&((*q)->l),ele);
96              else
97                  insertmain(&((*q)->r),ele);
98          }
99      }
100     return;
101 }
102
```

```
void BIN_TREE::display(int ch)
{
    switch(ch)
    {
    case 2:cout<<"\nPreorder Traversal : ";
        preorder(p);
        break;
    case 3:cout<<"\nInorder Traversal : ";
        inorder(p);
        break;
    case 4:cout<<"\nPostorder Traversal : ";
        postorder(p);
        break;
    }
}

void BIN_TREE::preorder(node *p)
{
    if(p!=NULL)
    {
        cout<<p->data<<" ";
        preorder(p->l);
        preorder(p->r);
    }
}

void BIN_TREE::inorder(node *p)
{
    if(p!=NULL)
    {
        inorder(p->l);
        cout<<p->data<<" ";
        inorder(p->r);
    }
}

void BIN_TREE::postorder(node *p)
{
    if(p!=NULL)
    {
        postorder(p->l);
        postorder(p->r);
        cout<<p->data<<" ";
    }
}
```

Output

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 14\Debug\Prg 14.exe
1. Insert
2. Preorder
3. Inorder
4. Postorder
5. Exit

Enter Selection : 1

Enter Element To Insert : 11
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 14\Debug\Prg 14.exe
1. Insert
2. Preorder
3. Inorder
4. Postorder
5. Exit

Enter Selection : 2

Preorder Traversal : 65 46 11 56 98 66
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 14\Debug\Prg 14.exe
1. Insert
2. Preorder
3. Inorder
4. Postorder
5. Exit

Enter Selection : 3

Inorder Traversal : 11 46 56 65 66 98
```

```
C:\Users\Akash\Desktop\Lab 3rd Sen\Prg 14\Debug\Prg 14.exe
1. Insert
2. Preorder
3. Inorder
4. Postorder
5. Exit

Enter Selection : 4

Postorder Traversal : 11 56 46 66 98 65
```