Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Fundamentals Of Data Structures

Assignment no 1

Problem statement->

Represent matrix using two dimensional arrays and perform following operations without pointers:

i. Addition

ii. Multiplication

iii.Transpose

Code->

```
#include <iostream>

using namespace std;

int main()

{

int r,c;

cout<<"enter the no  of rows in matrix";

cin>>r;

cout<<"enter the no  of columns in matrix";

cin>>c;

int a[r][c];

int b[r][c];

int add[r][c];
```

```
int mult[r][c];
cout<<"enter the elements of first matrix";
for(int i=0;i<r;i++)
{
  for(int j=0;j<c;j++)
  {
     cin>>a[i][j];
  }
}
   cout<<"enter the elements of second  matrix";
for(int i=0;i<r;i++)
{
  for(int j=0;j<c;j++)
  {
      cin>>b[i][j];
  }
}
for(int i=0;i<r;i++)
{
  for(int j=0;j<c;j++)
  {
       add[i][j]=a[i][j]+ b[i][j];
       mult[i][j]=0;
```

```
  }
}
   cout<<"enter the elements of addition matrix is ";
for(int i=0;i<r;i++)
{
    cout<<endl;
  for(int j=0;j<c;j++)
  {
      cout<<add[i][j]<<"\t\t";
  }
}
for(int i=0;i<r;i++)
{
    for(int j=0;j<c;j++)
    {
        for(int k=0;k<c;k++)
        {
            mult[i][j]+=a[i][k]*b[k][j];
        }
    }
}
 cout<<" the elements of multiplication matrix is "<<endl<<endl;
for(int i=0;i<r;i++)
```

```
{
    cout<<endl;
  for(int j=0;j<c;j++)
  {
      cout<<mult[i][j]<<"\t\t";
  }

}
return 0;
}
```

Output->

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
enter the no  of rows in matrix3
enter the no  of columns in matrix3
enter the elements of first matrix1
1
1
1
1
1
1
1
1
enter the elements of second  matrix2
2
2
2
2
2
2
2
2
enter the elements of addition matrix is
3               3               3
3               3               3
3               3               3               the elements of multiplication matrix is


6               6               6
6               6               6
6               6               6
Process returned 0 (0x0)   execution time : 12.319 s
Press any key to continue.
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Fundamentals Of Data Structures

Assignment no 2

Problem statement->

Represent matrix using two dimensional arrays and perform following operations without pointers:

i.Saddle Point

ii.Upper and Lower triangular matrix

CODE->

```
#include<iostream>
using namespace std;
int main()
{
int x,y,i=0,j=0,min1=0,count1 ,max1,p=0,m=0,k,ch;
int  arr[20][20],ul[20][20];
cout<<"\n\n enter the number of rows and columns";
cin>>x>>y;
cout<<"enter the elements mmatrix ";
 for(i=0;i<x;i++)
  {
    for(j=0;j<y;j++)
          {
          cin>>arr[i][j];

          }
  }

cout<<"the given matrix is"<<endl;
  for(i=0;i<x;i++)
```

```
 {
    cout<<endl;
  for(j=0;j<y;j++)
       {
       cout<<arr[i][j];
       cout<<"\t\t";

       }
 }


for(i=0;i<x;i++)
 {

    min1=arr[i][0];
        for(j=0;j<y;j++)
        {
                        if(arr[i][j]<min1)
                        {
                        min1=arr[i][j];
                        p=j;
                        }
         }

max1=min1;
count1=0;
  for(k=0;k<y;k++)
   {

                if(max1>arr[k][p])
     {
                  count1++;
     }
   }
                if(count1==k-1)
```

```
        {
                cout<<"\n\t the sadal point is ::";
            cout<<max1<<endl;
            m++;
        }
}
if(m==0)
{
   cout<<"\n\n\t";
cout<<"no sadal point !!!!";
}




do
{
cout<<endl<<endl;
cout<<"1.lower 2.upper 0.exit";
cin>>ch;


for(i=0;i<x;i++)
  {
    for(j=0;j<y;j++)
          {
           ul[i][j]=arr[i][j];

          }
  }

switch(ch)
{
```

```
case 1:
    cout<<"the lower trangular matrix is";
    for(i=0;i<x;i++)
  {
    for(j=0;j<y;j++)
          {
          if(j>i)
        ul[i][j]=0;

          }
  }

    break;
case 2:
    cout<<"the upper trangular matrix is";
    for(i=0;i<x;i++)
  {
    for(j=0;j<y;j++)
          {
              if(j<i)
        ul[i][j]=0;

          }
  }


 break;

}

for(i=0;i<x;i++)
  { cout<<endl;
    for(j=0;j<y;j++)
          {
          cout<<ul[i][j]<<"\t\t";
```

```
        }
  }
}while(ch!=0);
return 0;
}
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Output->

```
 enter the number of rows and columns3
3
enter the elements mmatrix 1
2
3
4
5
6
7
8
9
the given matrix is

1               2               3
4               5               6
7               8               9
         the sadal point is ::7


1.lower 2.upper 0.exit1
the lower trangular matrix is
1               0               0
4               5               0
7               8               9

1.lower 2.upper 0.exit2
the upper trangular matrix is
1               2               3
0               5               6
0               0               9

1.lower 2.upper 0.exit
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Fundamentals Of Data Structures

Assignment no 3

Problem statement->

Write a menu driven program in C++ for the following operations on Singly Linked List (SLL) of student data with the fields: PRN, Name, Branch, Semester, Cell Number a. Create a SLL of N Students b. Perform Insertion c. Displaythe SLL and count the number of nodes in it

Code->

```
#include<iostream>
#include<string.h>
using namespace std;
typedef struct linklist
{
char name[10];
int sem;
char branch[10];
char no[10];
char prn[10];
struct linklist *next;
}node;

node*getdata()
{
node *record;
```

```
record=new node;
cout<<"enter the semester in which you are";
cin>>record->sem;
cout<<"enter the name";
cin>>record->name;
cout<<"enter the branch name";
cin>>record->branch;
cout<<"enter the PRN";
cin>>record->prn;
cout<<"enter the contact no";
cin>>record->no;

record->next=NULL;
return record;
}

   // INSERT AT END
node* insert_end(node *first)
{
node *move,*newnode;
newnode=getdata();
if(first==NULL)
{
```

```
first=newnode;
cout<<"head is created!!!";
}
else
{
move=first;
while(move->next!=NULL)
move=move->next;
move->next=newnode;
}
return first;
}

// INSETR AT POSITION
node *insert_position(node *first)
{
int pos,count=1;
node *move,*newnode;
move=first;
cout<<" enter the position at which you wnat to insert";
cin>>pos;
newnode=getdata();
move=first;
```

```
while(count!=pos-1)
{
move=move->next;
count++;
}
newnode->next=move->next;
move->next=newnode;
return first;
}

// INSERT AT BEGINING
node *insert_beg(node *first)
{
node *move,*newnode;
newnode=getdata();
newnode->next=first;
first=newnode;
return first;
}
// DISPLAY LINKLIST
void display(node* first)
{
node *move;
```

```
move=first;
while(move!=NULL)
{
cout<<"\n\t"<<"branch ::      "<<move->branch<<"\n\t"<<"name::    "<<move->name<<"\n\t"<<"prn::  "<<move->prn<<"\n\t";
cout<<"semester::  "<<move->sem<<"\n\t";
cout<<"contact no::  "<<move->no<<"\n\t";

cout<<"\n\t";
cout<<"**********";
cout<<"\n\t";
move=move->next;
}
}
void count(node* first)
{
Int count;
node *move;
move=first;
while(move!=NULL)
{
move=move->next;
count++;
```

```
}
Cout<<count<<"no of nodes";
}

 int main()
{
node *first=NULL;
int no,ch,ele;

do
{
cout<<"\n\t 1.insert node at end  \n\t 2.insert node at position \n\t3. insert node at begining \n\t 4.display \n\t 5.count 0.exit
cout<<"\n\t enter your choise";
cin>>ch;
switch(ch)
{
case 1:
 first=insert_end(first);
break;
case 2:
 first=insert_position(first);
 break;
```

```
case 3:
first=insert_beg(first);
break;
case 4:
display(first);
break;
case 5:
count(first);
break;
}
}
while(ch!=0);
return 0;
}
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
          1.insert node at end
          2.insert node at position
         3. insert node at begining
          4.display
          5.count 0.exit
          enter your choise1

enter the semester in which you are  3
enter the name  shivam
enter the branch name  computer
enter the PRN  1234
enter the contact no  8911995001

head is created!!!


          1.insert node at end
          2.insert node at position
         3. insert node at begining
          4.display
          5.count 0.exit
          enter your choise3

enter the semester in which you are  3
enter the name  yash
enter the branch name  computer
enter the PRN  4321
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
        1.insert node at end
        2.insert node at position
       3. insert node at begining
        4.display
        5.count 0.exit
        enter your choise2
 enter the position at which you wnat to insert2

enter the semester in which you are  3
enter the name  saumitra
enter the branch name  computer
enter the PRN  431401
enter the contact no  907766553




        1.insert node at end
        2.insert node at position
       3. insert node at begining
        4.display
        5.count 0.exit
        enter your choise4

      branch ::      computer
      name::    yash
      prn::  4321
      semester::  3
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
branch ::       computer
name::    saumitra
prn::  431401
semester::  3
contact no::  907766553

**********

branch ::       computer
name::    shivam
prn::
semester::  3
contact no::  8911995001

**********



 1.insert node at end
 2.insert node at position
3. insert node at begining
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Fundamentals Of Data Structures

Assignment no 04

Problem statement->

Write a menu driven program in C++ for the following operations on Singly Linked List (SLL) of student data with the fields: PRN, Name, Branch, Semester, Cell Number a. Create a SLL b. Search a node c. Deletion of node

Code->

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

typedef struct linklist
{
char name[10];
char branch[10];
char no[10];
char prn[10];
int sem;
struct linklist *next;
}node;

node*getdata()

{

node *record;

record=new node;

cout<<"enter the semester in which you are";

cin>>record->sem;
```

```
cout<<"enter the name";
cin>>record->name;
cout<<"enter the branch name";
cin>>record->branch;
cout<<"enter the PRN";
cin>>record->prn;
cout<<"enter the contact no";
cin>>record->no;

record->next=NULL;
return record;
}


// INSERT AT END
node* insert_end(node *first)
{
node *move,*newnode;
newnode=getdata();
if(head==NULL)
{
head=newnode;
cout<<"head is created!!!";
```

```
}
else
{
move=first;
while(move->next!=NULL)
move=move->next;
move->next=newnode;
}
return first;
}

// DISPLAY LINKLIST
void display(node* first)
{
node *move;
move= first;
while(move!=NULL)
{
cout<<"\n\t"<<"branch ::      "<<move->branch<<"\n\t"<<"name::    "<<move-
>name<<"\n\t"<<"prn::  "<<move->prn<<"\n\t";
cout<<"semester::  "<<move->sem<<"\n\t";
cout<<"contact no::  "<<move->no<<"\n\t";
```

```
cout<<"\n\t";

cout<<"* * * * * * * * * *";

cout<<"\n\t";

move=move->next;

}

}

// delete at beginning
node *delete_beg(node *first)
{
node *move;
move=first;
head=head->next;
move->next=NULL;

cout<<"deleted record is";

cout<<"\n\t"<<"branch ::      "<<move->branch<<"\n\t"<<"name::    "<<move->name<<"\n\t"<<"prn::  "<<move->prn<<"\n\t";

cout<<"semester::  "<<move->sem<<"\n\t";

cout<<"contact no::  "<<move->no<<"\n\t";

cout<<"\n\t";
cout<<"**********";
cout<<"\n\t";

delete move;
return first;
    }

        // DELETE AT END
```

```
// DELETE AT END
node *delete_end(node * first)
{
 int count=0,num=0;
 node *move,*temp;
  move=head;
if(head==NULL)
 {
cout<<"list is empty";
 }

 else
 {
  while(move->next!=NULL)
   {
    temp=move;
    move=move->next;
    }
    temp->next=NULL;

    cout<<"the deleted record is";
       1) cout<<"\n\t"<<"branch ::      "<<move->branch<<"\n\t"<<"name::
          "<<move->name<<"\n\t"<<"prn::  "<<move->prn<<"\n\t";
       2) cout<<"semester::  "<<move->sem<<"\n\t";
       3) cout<<"contact no::  "<<move->no<<"\n\t";
       4)
cout<<"\n\t";
cout<<"**********";
cout<<"\n\t";
   delete move;
  }

  return first;
}
          // search
  void search(node * first)
```

```
 {
 node *move;
move= first;
 int empid;
 cout<<"enter the name";
 cin>> name;

 while(move!=NULL && move-> name!= name)
 move=move->next;

 if(move==NULL)
 cout<<" \n\n Record not found";

 else
  cout<<"\n\n\t the record belonging to that name is";
     cout<<"\n\t"<<"branch ::      "<<move->branch<<"\n\t"<<"name::    "<<move-
     >name<<"\n\t"<<"prn::  "<<move->prn<<"\n\t";

     cout<<"semester::  "<<move->sem<<"\n\t";

     cout<<"contact no::  "<<move->no<<"\n\t";

     1)
cout<<"\n\t";
cout<<"**********";
cout<<"\n\t";
 }

 // delete at given position
 node * Delete_position(node * first)
 {
 int pos,count=1;
 cout<<"enter the which you want to delete ";
 cin>>pos;

 node*move,*temp;
```

```
move=head;

while(count!=pos)
{
temp=move;
move=move->next;
count++;
}
temp->next=move->next;
cout<<"deleted record is";

    cout<<"\n\t"<<"branch ::      "<<move->branch<<"\n\t"<<"name::    "<<move-
    >name<<"\n\t"<<"prn::  "<<move->prn<<"\n\t";

    cout<<"semester::  "<<move->sem<<"\n\t";

    cout<<"contact no::  "<<move->no<<"\n\t";

cout<<"\n\t";
cout<<"* * * * * * * * * *";
cout<<"\n\t";
delete move;
return first;
}
```

Output->

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
         1.delete_end
         2.Delete_position
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise5

enter the semester in which you are  2
enter the name  shivam
enter the branch name  computer
enter the PRN  321
enter the contact no  9088776654
head is created!!!


         1.delete_end
         2.Delete_position
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise5

enter the semester in which you are  3
enter the name  yash
enter the branch name  comp
enter the PRN  6789
enter the contact no  1234567



         1.delete_end
         2.Delete_position
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise5

enter the semester in which you are  5
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
         2.Delete_position
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise5

enter the semester in which you are  4
enter the name  ankit
enter the branch name  civil
enter the PRN  5678
enter the contact no  123456789



         1.delete_end
         2.Delete_position
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise5

enter the semester in which you are  6
enter the name  aniket
enter the branch name  mech
enter the PRN  786
enter the contact no  333334644564



         1.delete_end
         2.Delete_position
        3.delete_beg
         4.display
```

```
branch ::       computer
name::    shivam
prn::
semester::  2
contact no::  9088776654

**********

branch ::       comp
name::    yash
prn::  6789
semester::  3
contact no::  1234567

**********

branch ::       entc
name::    shardual
prn::  556
semester::  5
contact no::  3333444455556

**********

branch ::       civil
name::    ankit
prn::  5678
semester::  4
contact no::  123456789

**********

branch ::       mech
name::    aniket
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
         1.delete_end
         2.Delete_position
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise1
the deleted record is
        branch ::       mech
        name::    aniket
        prn::  64
        semester::  6
        contact no::  333334644564

        **********



         1.delete_end
         2.Delete_position
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise3
deleted record is
        branch ::       computer
        name::    shivam
        prn::
        semester::  2
        contact no::  9088776654

        **********



         1.delete_end
         2.Delete_position
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise2
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise2
enter the which you want to delete 2
deleted record is
        branch ::       entc
        name::    shardual
        prn::  556
        semester::  5
        contact no::  3333444455556

        **********



         1.delete_end
         2.Delete_position
        3.delete_beg
         4.display
         5.create 0.exit
         enter your choise4

        branch ::       comp
        name::    yash
        prn::  6789
        semester::  3
        contact no::  1234567

        **********

        branch ::       civil
        name::    ankit
        prn::  5678
        semester::  4
        contact no::  123456789
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Fundamentals Of Data Structure
Assignment – 05

1. Aim:

Implement a menu driven Program in C++ for the following operations on stack using ARRAY such as

a. Push an element on to Stack
b. Pop an element
c. Demonstrate overflow situations on Stack
d. Display stack.

Code->

```
#include<iostream>
using namespace std;
#define MAXSIZE 50
typedef struct STACK
{
int arr[MAXSIZE];
int top;
}stack;
int Isfull(stack *st)
{
	if(st->top==MAXSIZE-1)
	return 1;
	else
	return 0;
}

int Isempty(stack *st)
{
if(st->top == -1)
{
return 1;
}
else
{
```

```
return 0;
}
}
void push(stack *st,int ele)
 {
                if(Isfull(st))
        cout<<"stack is full";
else
                        {
                        st->top++;
                        st->arr[st->top]=ele;
                        }
  }

void  pop(stack *st)
 {
  int ele;
                if (Isempty(st))
                {
                cout<<"stack is empty "<<endl;
                }
           else
                        {
                if(ele=st->arr[st->top];
                        st->top--;
      cout<<"deleted element is"<<ele<<endl;
  }
 }
void display(stack *st)
{

  cout<<"elemnts of stack are"<<endl;
                for(int i=st->top;i>=0;i--)
            cout<<st->arr[i]<<endl;
```

```
}

int main()
{
int ch,no;
stack st;
st.top=-1;
```

```
1.push 2.pop 3.display 0.exit
1
 enter ele
11
1.push 2.pop 3.display 0.exit
1
 enter ele
12
1.push 2.pop 3.display 0.exit
1
 enter ele
13
1.push 2.pop 3.display 0.exit
3
elemnts of stack are
13
12
11
1.push 2.pop 3.display 0.exit
2
deleted element is13
1.push 2.pop 3.display 0.exit
2
deleted element is12
1.push 2.pop 3.display 0.exit
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Fundamentals Of Data Structure
Assignment – 06

1. Aim:

Implement a menu driven Program in C++ for the following operations on stack using Linked List (SLL) such as

a. Push an element on to Stack
b. Pop an element
c. Demonstrate overflow situations on Stack
d. Display stack.

Code->

```
#include<bits/stdc++.h>
using namespace std;
typedef struct STACK
{
int data;
struct STACK *next;
}node;


node *accept()
{
node *newnode;
newnode= new node;
cout<<"enter the element"<<endl;
cin>>newnode->data;
newnode->next=NULL;

return newnode;

}
int Isempty(node *top)
{
if(top == NULL)
{
  return 1;
```

```
}
else
{
return 0;
}

}

node *push(node *top)
{
node *newnode;
newnode=accept();

if(top==NULL)
{
top=newnode;

}
else
{
newnode->next=top; //the main line
top=newnode;//yhe main line

}
return top;
}

node* pop(node *top)
 {
node*move;
move=top;
if(Isempty(top))
        {
        cout<<"the stack is empty"<<endl;
        }
else
        {
                top=move->next;/// the main line
```

```
                cout<<"the deleted data is"<<endl<<move->data;
                delete(move);/// the main line
        }
return top;
 }


void display(node *top)
 {

if(Isempty(top))
        {
        cout<<"stack is empty"<<endl;
        }
else
                {
                        node *move;
                        move=top;

                while(move!=NULL)
                        {
                        cout<<move->data<<endl;
                        move=move->next;

                        }
                 }

}
int main()
{
int ch;
node *top;
top=NULL;
do
 {
cout<<endl<<"1.push \n2.pop\n 3.display\n 0.exit\n";
cin>>ch;
   cout<<endl;
```

```
switch(ch)
                {
                case 1:
                top=push(top);
                break;
                case 2:
                top=pop(top);
                break;

                case 3:
                display(top);
                break;

        default:
           cout<<"enter the correct option";
                }

}while(ch!=0);
return 0;
}
```

Output->

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
1.push
2.pop
 3.display
 0.exit
1

enter the element
11

1.push
2.pop
 3.display
 0.exit
1

enter the element
12

1.push
2.pop
 3.display
 0.exit
1

enter the element
13

1.push
2.pop
 3.display
 0.exit
3
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
1.push
2.pop
 3.display
 0.exit
3

13
12
11

1.push
2.pop
 3.display
 0.exit
2

the deleted data is
13
1.push
2.pop
 3.display
 0.exit
2

the deleted data is
12
1.push
2.pop
 3.display
 0.exit
2

the deleted data is
11
1.push
2.pop
 3.display
 0.exit
2
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
the deleted data is
11
1.push
2.pop
 3.display
 0.exit
2

the stack is empty

1.push
2.pop
 3.display
 0.exit
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Fundamentals Of Data Structures

Assignment no 07

Code->

```
#include<iostream>
#include<string.h>
#define ERROR -9999
using namespace std;
typedef struct Linklist
{
        char data[50];
        struct Linklist *next;
}Queue;

Queue* CreateNode(char ele[])
{
        Queue *record;
        record= new Queue;
        strcpy(record->data,ele);
```

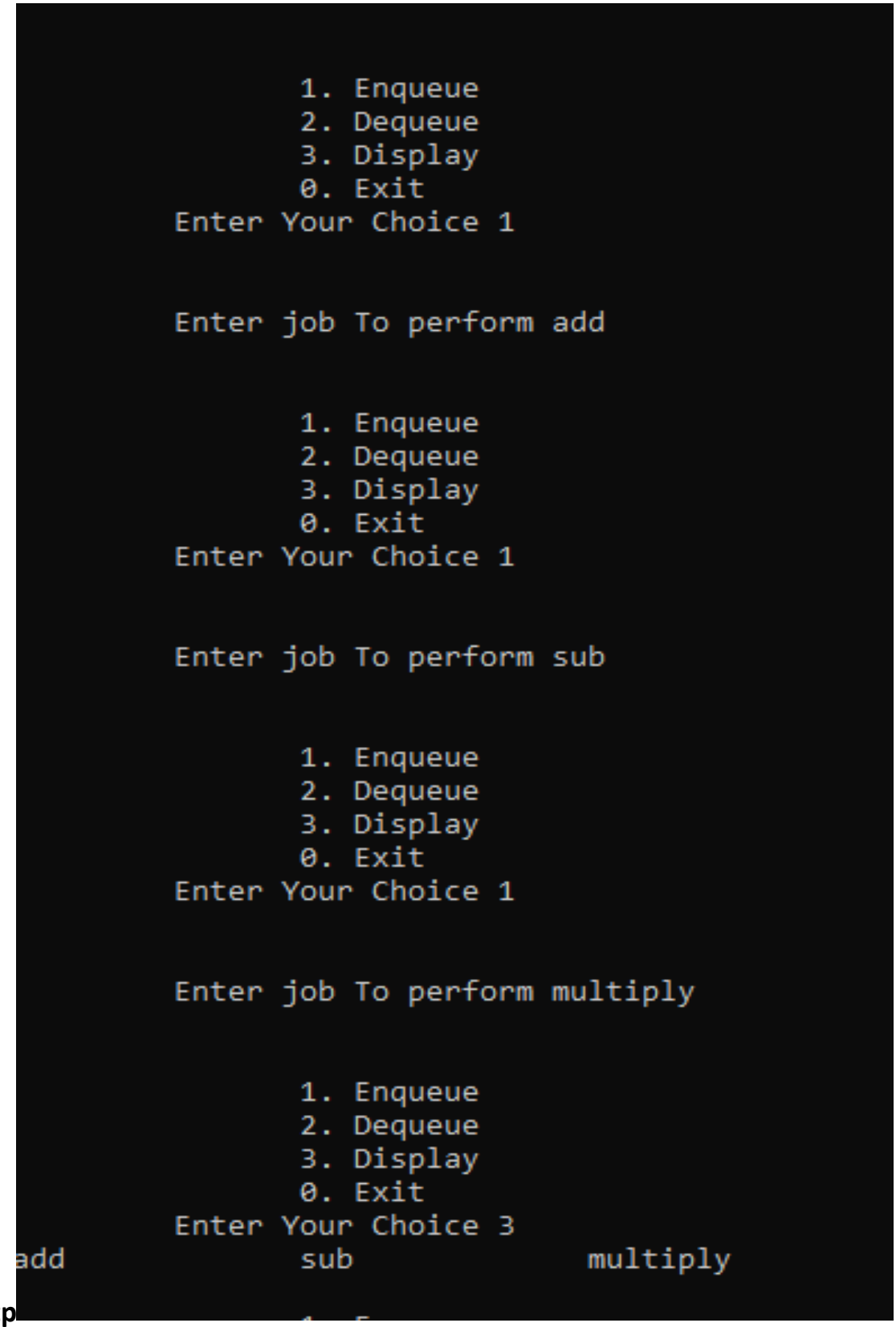
```
        record->next=NULL;
        return record;
}
void EnQueue(Queue **r,Queue **f, char ele[])
{
        Queue *newnode=CreateNode(ele);
        if(*f==NULL)
        {
                *r=*f=newnode;
        }
        else
        {
                (*r)->next=newnode;
                *r=newnode;
        }
}
void DeQueue(Queue **r,Queue **f)
{
        char ele[50];
        Queue *del;
        if(*f==NULL)
                cout<<"\n\n\t Queue Is Empty !!! ";
        else
```

```
        {
                del=*f;
                *f=del->next;
                strcpy(ele,del->data);
                delete del;
        }
cout<<" \t\tjob completed."<<ele;
}
void Display(Queue *r,Queue *f)
{
   if(f ==NULL)
    cout<<"\t\tqueue is empty";
        else
        {
        Queue *move;
        move=f;

        while(move!=r)
        {
            cout<<move->data<<"\t\t";

                move=move->next;
        }
```

```
        if(move==r)
        cout<<move->data<<"\t\t";

}

}

int main()
{
        int ch;
    char ele[50];
        Queue *front=NULL,*rear=NULL;
        do
        {
                cout<<"\n\n\t\t1. Enqueue \
                        \n\t\t2. Dequeue\
                            \n\t\t3. Display\
                            \n\t\t0. Exit\
                            \n\t Enter Your Choice ";
                cin>>ch;
                switch(ch)
                {
                case 1:
```

```
                cout<<"\t\t\n\n\t Enter job To perform ";
                cin>>ele;
                EnQueue(&rear,&front,ele);
                break;
            case 2:
                DeQueue(&rear,&front);

                break;
            case 3:
                Display(rear,front);
                break;
            }
        }while(ch!=0);
        cout<<"\n\n\n";
return 0;
}
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020



tp

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
        1. Enqueue
        2. Dequeue
        3. Display
        0. Exit
Enter Your Choice 2
        job completed.add

        1. Enqueue
        2. Dequeue
        3. Display
        0. Exit
Enter Your Choice 2
        job completed.sub

        1. Enqueue
        2. Dequeue
        3. Display
        0. Exit
Enter Your Choice 2
        job completed.multiply

        1. Enqueue
        2. Dequeue
        3. Display
        0. Exit
Enter Your Choice 2


Queue Is Empty !!!              job completed.K

        1. Enqueue
        2. Dequeue
        3. Display
        0. Exit
Enter Your Choice
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Fundamentals of data structure

Assignment no 8

Problem statement->

In a hospital emergency room, arrange the patients according to severity of their problem even if they have been waiting longer. Implement it with the help of suitable data structure.

Code->

```
#include <iostream>

#include <cstdio>

#include <cstring>

#include <cstdlib>

using namespace std;



struct node

{

        int priority;

        char  info[100];

        struct node *link;

};
```

```
class Priority_Queue
{
    private:
        node *front;
    public:
        Priority_Queue()
        {
            front = NULL;
        }

        void insert(char item[100], int priority)
        {
            node *tmp, *q;
            tmp = new node;
           strcpy(tmp->info ,item);
            tmp->priority = priority;
            if (front == NULL || priority < front->priority)
            {
                tmp->link = front;
                front = tmp;
            }
```

```
    else
    {
        q = front;
        while (q->link != NULL && q->link->priority <= priority)
            q=q->link;
        tmp->link = q->link;
        q->link = tmp;
    }
}

void del()
{
    node *tmp;
    if(front == NULL)
        cout<<"\t\tQueue Underflow\n";
    else
    {
        tmp = front;
        cout<<"\t\t Deleted item is: "<<tmp->info<<endl;
        front = front->link;
        free(tmp);
```

```
        }
          cout<<endl;
    }

    void display()
    {
        node *ptr;
        ptr = front;
        if (front == NULL)
            cout<<"\t\t Queue is empty\n";
        else
        {    cout<<"\t\tQueue is :\n";
            cout<<"\t\t  Priority       Item\n";
            while(ptr != NULL)
            {
                cout<<"\t\t"<< ptr->priority<<"         "<<ptr->info<<endl;
                ptr = ptr->link;
            }
        }
    }
};
```

```
int main()
{


    cout<<endl<<endl;
    char item[100];
    int choice,  priority;
    Priority_Queue pq;
    do
    {
        cout<<endl;
        cout<<"\t\t1.Insert\n";
        cout<<"\t\t2.Delete\n";
        cout<<"\t\t3.Display\n";
        cout<<"\t\t4.Quit\n";
        cout<<"\t\t Enter your choice : ";
        cin>>choice;
        switch(choice)
        {
        case 1:
```

```
            cout<<"\t\t enter the disease of patient  : ";
            cin>>item;
            cout<<"\t\t Enter its priority : ";
            cin>>priority;
            pq.insert(item, priority);
            break;
        case 2:
            pq.del();
            break;
        case 3:
            pq.display();
            break;
        case 4:
            break;
        default :
            cout<<"Wrong choice\n";
        }
          cout<<endl;
    }
    while(choice != 4);
    return 0;}
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
1.Insert
2.Delete
3.Display
4.Quit
 Enter your choice : 1
 enter the disease of patient  : cough
 Enter its priority : 4


1.Insert
2.Delete
3.Display
4.Quit
 Enter your choice : 1
 enter the disease of patient  : accident
 Enter its priority : 1


1.Insert
2.Delete
3.Display
4.Quit
 Enter your choice : 1
 enter the disease of patient  : dengu
 Enter its priority : 2


1.Insert
2.Delete
3.Display
4.Quit
 Enter your choice : 1
 enter the disease of patient  : fever
 Enter its priority : 3


1.Insert
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
Queue is :
  Priority        Item
1           accident
2           dengu
3           fever
4           cough


1.Insert
2.Delete
3.Display
4.Quit
 Enter your choice : 2
 Deleted item is: accident



1.Insert
2.Delete
3.Display
4.Quit
 Enter your choice : 2
 Deleted item is: dengu



1.Insert
2.Delete
3.Display
4.Quit
 Enter your choice : 2
 Deleted item is: fever



1.Insert
2.Delete
3.Display
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

```
1.Insert
2.Delete
3.Display
4.Quit
 Enter your choice : 2
Queue Underflow



1.Insert
2.Delete
3.Display
4.Quit
 Enter your choice : 3
 Queue is empty


1.Insert
2.Delete
3.Display
4.Quit
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

FUNDAMENTALS OF DATA STRUCTURE

ASSIGNMENT 9

Problem statement->

Write C++ program to maintain club members, sort on roll numbers in ascending order.

CODE->

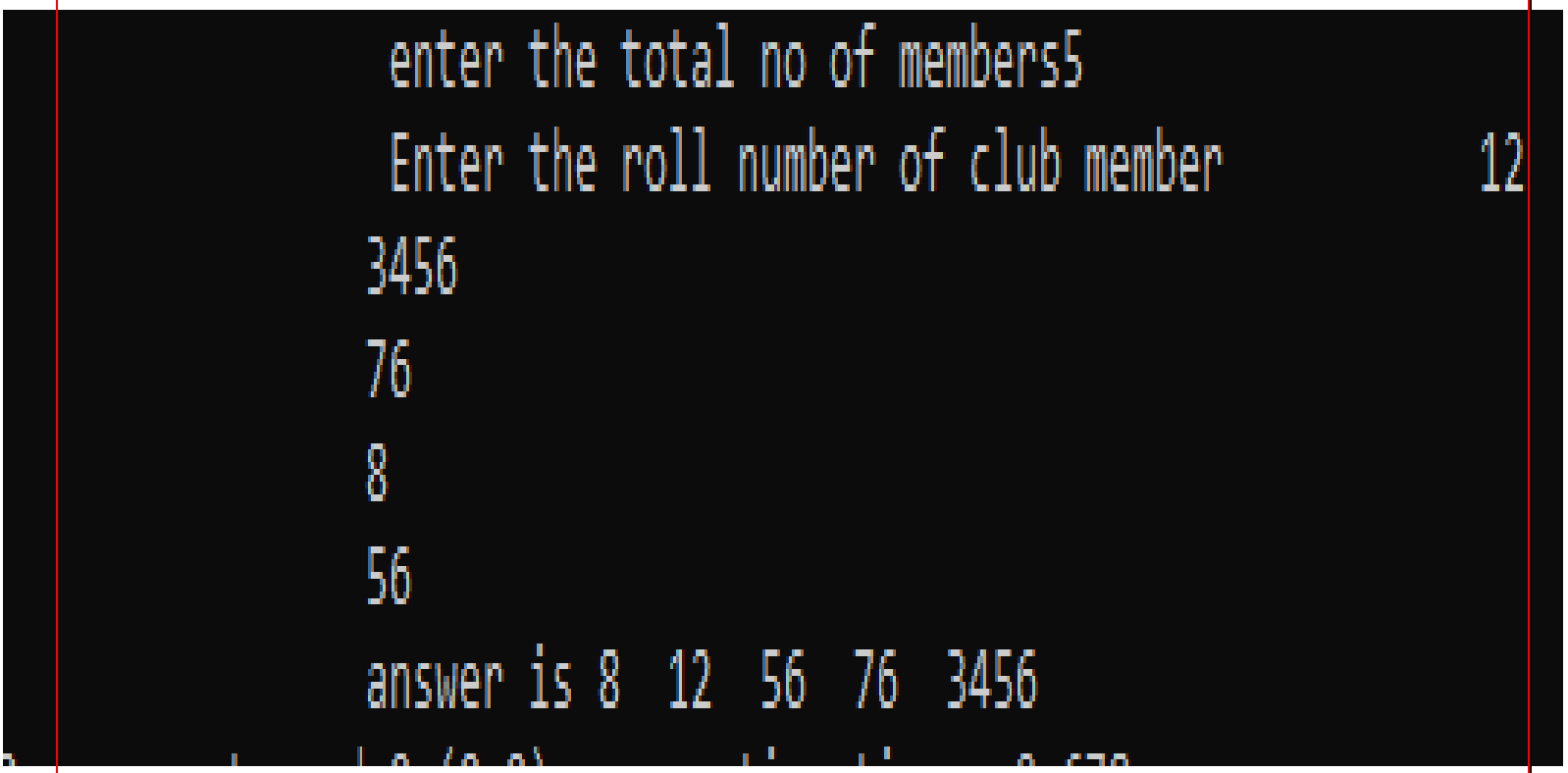
```
#include <iostream>
using namespace std;
 void insertsort(int arr[],int n)
 {                              //insertion sort function
  int i, key, j;
 for (i = 1; i < n; i++)
 {
    key = arr[i];
    j = i-1;

  while (j >= 0 && arr[j] > key)
    {
        arr[j+1] = arr[j];
        j = j-1;
    }
    arr[j+1] = key;
 }
```

```
    cout<<"\t\tanswer is ";
    for(i=0;i<n;i++)
       cout<<arr[i]<<"  ";

  }
int main()
{
   int no;
   cout<<"\t\t enter the total no of members";
   cin>>no;
   int a[no];

   cout<<"\t\t Enter the roll number of club member ";
   for(int i=0;i<no;i++)
   {
      cout<<"\t\t"; cin>>a[i];

   }

    insertsort(a,no);
   return 0;
```

```
}
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Output->



Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Fundamentals of data structure

Assignment no 10

Problem statement->

Write C++ program to implement Ternary Search for suitable application

Code->

```
#include <iostream>
using namespace std;
int ternary_search (int v[],int n, int left, int right, int x);
int main()
{
    cout<<"\n\n\t";
      int s;
cout<<"\t\t enter the number of club members  "<<endl;
 cout<<"\t\t \t\t ";cin>>s;
    int v[s];
    int x;
    cout<<"\t\t enter the number of roll of club ";
    cout<<endl;
    for(int i = 1; i <= s; i++)
    {
       cout<<"\t\t"; cin>>v[i-1];
    }
    cout << " \t\t Enter number for research:\n";
```

```
  cout<<"\t\t"; cin >> x;


    int left = s/3;
    int right = (s/3)*2;


    if(ternary_search(v,s,left-1,right-1,x) == -1)
    {
         cout<<" \t\t Number does not exist in array.\n";
    }
    else
    {
        cout<<"\t\t The index is:"<<ternary_search(v,s,left-1,right-1,x)+1<<"\n";
    }
    return 0;
}
int ternary_search (int v[],int n, int left, int right, int x)
{
    if(left < 0 || right > n-1 || left > right)
    {
        return -1;
    }
    if(x == v[left])
    {
```

```
        return left;
    }
    if(x == v[right])
    {
        return right;
    }
    if(x < v[left])
    {
        return ternary_search(v,n,left-1,right,x);
    }
    if (x > v[left] && x < v[right])
    {
        return ternary_search(v,n,left+1,right-1,x);
    }
    if(x > v[right])
    {
        return ternary_search(v,n,left,right+1,x);
    }
}
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Output->

```
          enter the number of club members
                    5
 enter the number of roll of club
21
31
41
51
61
 Enter number for research:
51
 The index is:4
```

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020

Name: shivam parve **Batch**-> B1 **Grno**->17u113 **Rollno**->222020