



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence



FINAL LAB REPORT

DATA STRUCTURE LAB



Submitted to:

Name: Most. Jannatul Ferdous

Department of CSE

Submitted by:

Name: Sadia Tasnim

Id : 20211103012

Intake: 46 Section: 1 (day)



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence

COURSE TITLE: DATA STRUCTURE

COURSE CODE: CSE232

LAB 1

Submitted to:

Name: Most. Jannatul Ferdous

Assistant professor

Department of cse

Bangladesh university of business and technology (BUBT)

Submitted by:

Name: Sadia Tasnim

Id: 20211103012

Intake: 46 Section: 1(day)

Bangladesh university of business and technology (BUBT)

Solution of Question No 01:

```
#include<stdio.h>
int add(int p, int q)
{
    return p+q;
}
int sub(int p, int q)
{
    return p-q;
}
int max (int p, int q)
{
    if(p>q)
        return p;
    else
        return q;
}
int min(int p, int q)
{
    if(p<q)
        return p;
    else
        return q;
}

int main()
{
    int x,y;
    int sum, sbtruc, maxi, mini;

    printf("Enter the value of a & b: ");
    scanf("%d%d",&x,&y);

    sum= add(x,y);
    printf("Addition:  %d \n",sum);

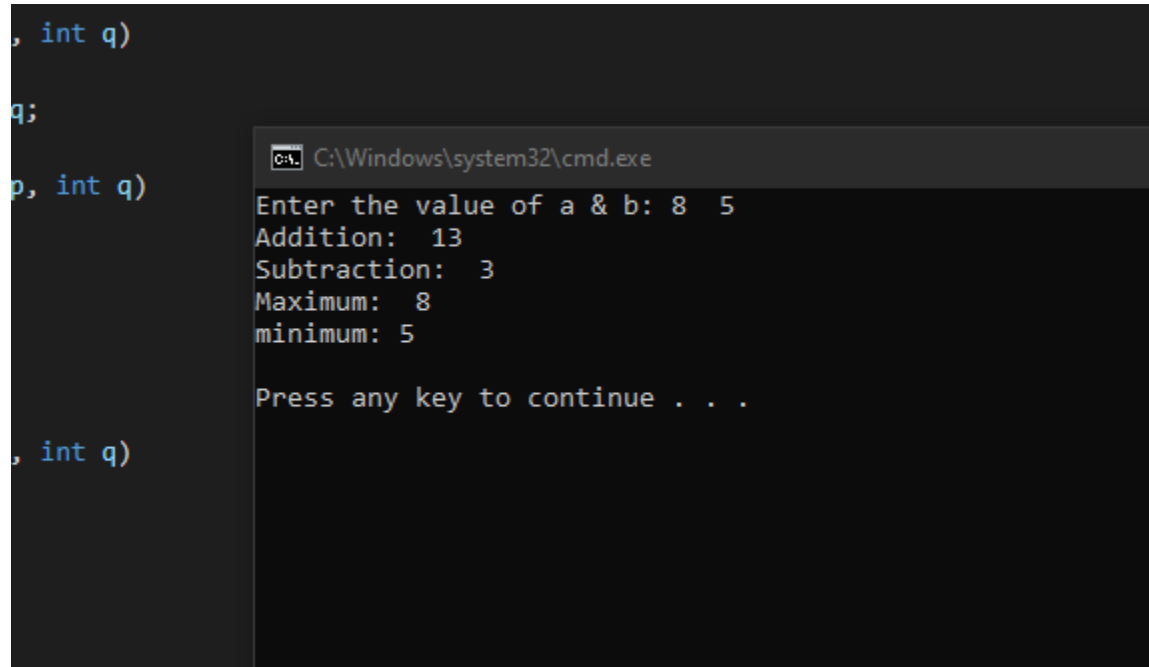
    sbtruc= sub(x,y);
    printf("Subtraction:  %d \n",sbtruc);

    maxi= max(x,y);
    printf("Maximum:  %d \n",maxi);
    mini= min(x,y);
```

```
printf("minimum: %d \n", mini);
```

```
}
```

Output:

A screenshot of a C++ program's output in a Windows command prompt. The program calculates the sum, difference, maximum, and minimum of two numbers, 8 and 5. The output is displayed in a black window with white text. The title bar of the window reads 'C:\Windows\system32\cmd.exe'. The program's output is as follows:

```
Enter the value of a & b: 8 5
Addition: 13
Subtraction: 3
Maximum: 8
minimum: 5

Press any key to continue . . .
```

Solution of Question No 02:

```
#include<stdio.h>
```

```
float area(float x)
```

```
{
```

```
float s;
```

```
s=3.1416*x*x;
```

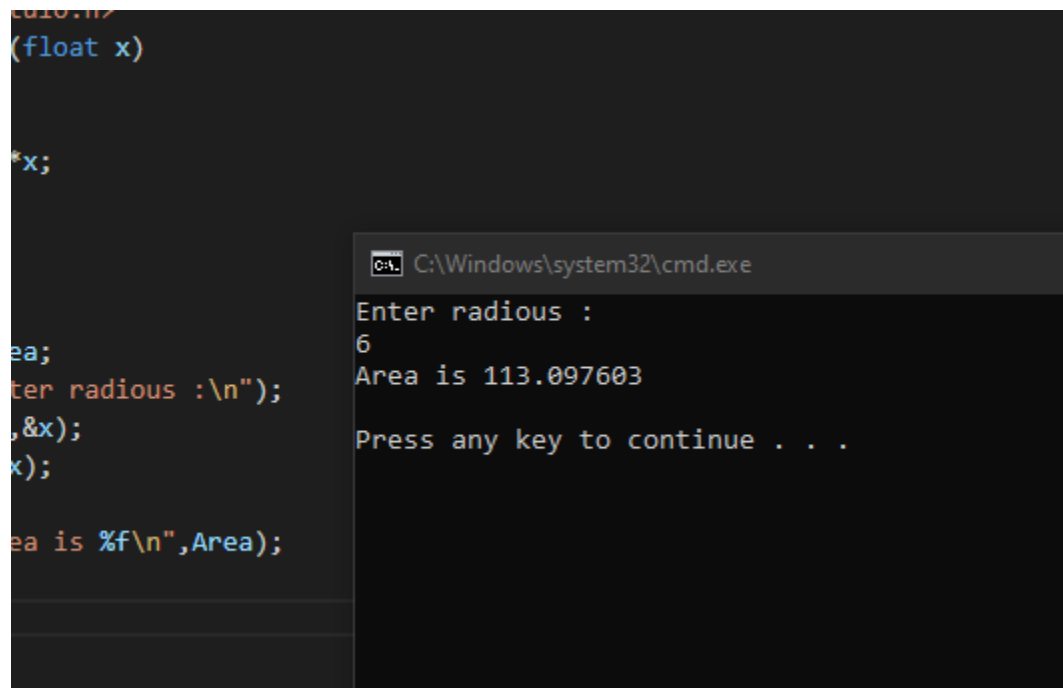
```
return s;
```

```
}
```

```
int main()
{
float x,Area;
printf("Enter radius : \n");
scanf("%f",&x);
Area=area(x);

printf("Area is %f\n",Area);
return 0;
}
```

Output:



The image shows a screenshot of a C++ program being executed. On the left, a portion of the source code is visible, showing the `main` function and the `printf` and `scanf` statements. On the right, a command prompt window titled `C:\Windows\system32\cmd.exe` displays the program's output. The output shows the prompt `Enter radius :`, the user input `6`, the calculated area `Area is 113.097603`, and the prompt `Press any key to continue . . .`.

```
C:\Windows\system32\cmd.exe
Enter radius :
6
Area is 113.097603
Press any key to continue . . .
```

Solution of Question No 03:

```
#include<stdio.h>

void print_array(int a[],int n)
{
    a[3]=55;
}

int main()
{
    int array[100];
    int n,i;

    printf("Enter n :");
    scanf("%d",&n);

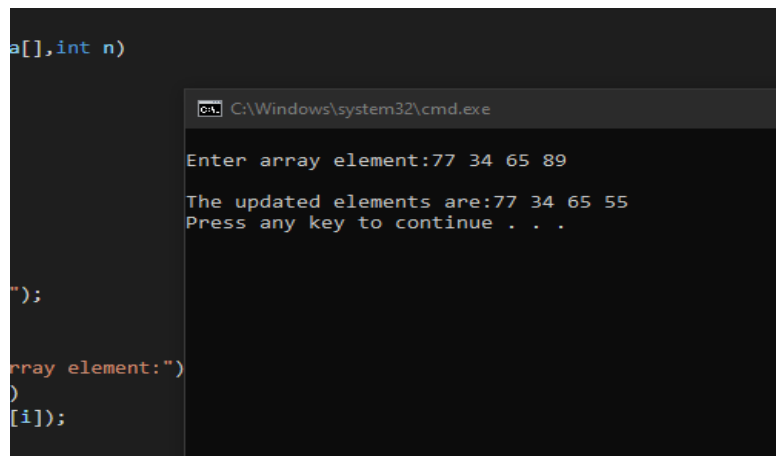
    printf("\nEnter array element:");
    for(i=0; i<n; i++)
        scanf("%d",&array[i]);

    print_array(array,n);

    printf("\nThe updated elements are:");
    for(i=0; i<n; i++)
        printf("%d ",array[i]);

}
```

Output:

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window shows the execution of a C program. The first prompt is "Enter array element:", followed by the input "77 34 65 89". The next line of output is "The updated elements are:", followed by the output "77 34 65 55". The final line of output is "Press any key to continue . . .". The background of the command prompt is dark, and the text is white. The program's source code is visible in the background, showing the same code as the previous block, with the line "a[3]=55;" highlighted in red.

```
a[],int n)

C:\Windows\system32\cmd.exe

Enter array element:77 34 65 89

The updated elements are:77 34 65 55
Press any key to continue . . .

");

array element:")
)
[i]);
```

Answer to the solution no 4:

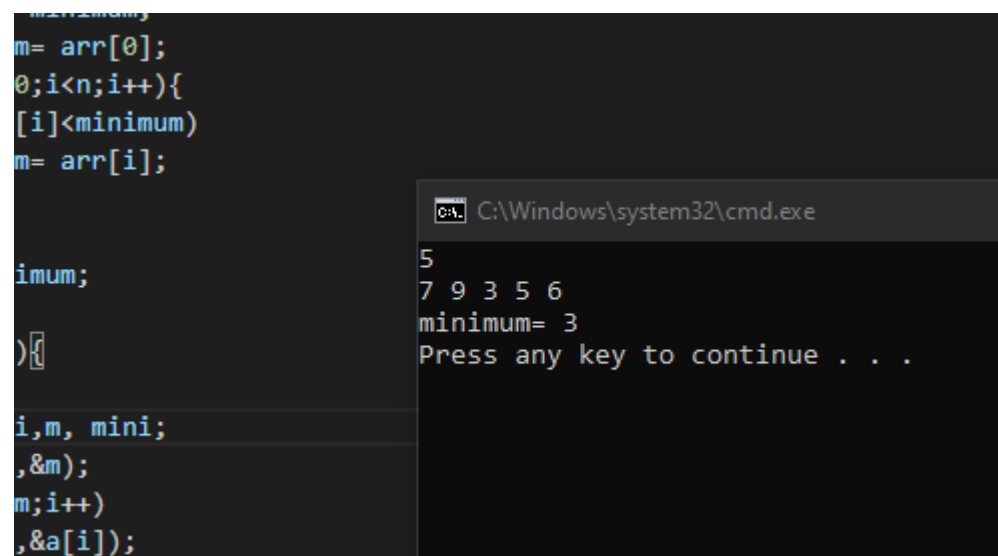
```
#include<stdio.h>

int minimum (int arr[],int n)
{
    int i, minimum;
    minimum= arr[0];
    for(i=0;i<n;i++){
        if(arr[i]<minimum)
            minimum= arr[i];
    }
    return minimum;
}

int main (){

    int a[10],i,m, mini;
    scanf("%d",&m);
    for(i=0;i<m;i++)
        scanf("%d",&a[i]);
    mini=minimum(a,m);
    printf("minimum= %d",mini);
}
```

Output:



The image shows a screenshot of a C program being executed in a Windows command prompt. The code on the left is a function to find the minimum value in an array. The output on the right shows the program running, with the user inputting 5 for the number of elements, followed by the array elements 7, 9, 3, 5, and 6. The program then outputs 'minimum= 3' and prompts the user to press any key to continue.

```
minimum;
m= arr[0];
0;i<n;i++){
[i]<minimum)
m= arr[i];

imum;
){
i,m, mini;
,&m);
m;i++)
,&a[i]);
```

C:\Windows\system32\cmd.exe

5
7 9 3 5 6
minimum= 3
Press any key to continue . . .



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence

COURSE TITLE: data structure

COURSE CODE: cse 232

Lab report 2

Submitted to:

Name: Most. Jannatul Ferdous

Assistant professor

Department of cse

Bangladesh university of business and technology (BUBT)

Submitted by:

Name: Sadia Tasnim

Id : 20211103012

Intake: 46 Section: 1 (day)

Bangladesh university of business and technology (BUB)

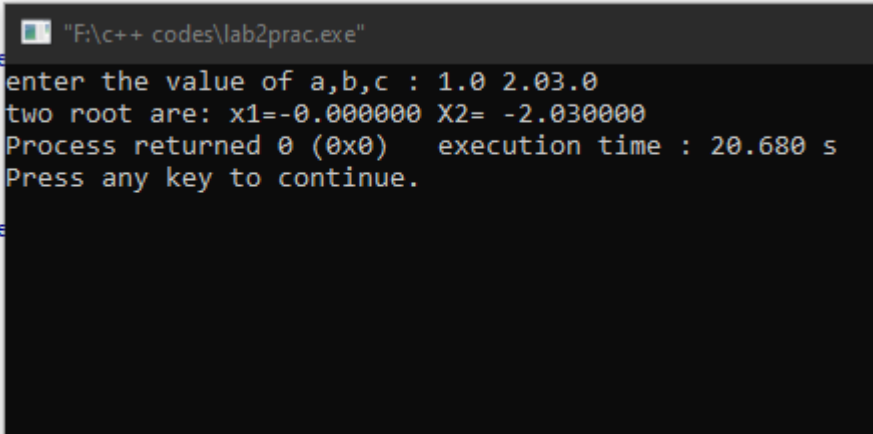
Answer to the question no 1

```
#include<stdio.h>
#include<math.h>
int main()
{
    float a,b,c,d;
    float x1,x2;
    printf("enter the value of a,b,c : ");
    scanf("%f%f%f",&a,&b,&c);
    d=b*b-4*a*c;
    if(d>0){
        x1=(-b+sqrt(d))/(2*a);
        x2=(-b-sqrt(d))/(2*a);
        printf("two root are: x1=%f X2= %f",x1,x2);

    }
    else if(d==0){
        x1=(float)-b/(2*a);
        printf("unique solution x1=%f",x1);

    }
    else {
        printf("no real solution");
    }
}
```

Out put:



The screenshot shows a Windows command prompt window titled "F:\c++ codes\lab2prac.exe". The program prompts the user to enter values for a, b, and c. The user enters "1.0 2.03.0". The program outputs "two root are: x1=-0.000000 X2= -2.030000". Below this, it shows "Process returned 0 (0x0) execution time : 20.680 s" and "Press any key to continue." The user has pressed a key, and the cursor is visible on the next line.

```
}
else
    enter the value of a,b,c : 1.0 2.03.0
    two root are: x1=-0.000000 X2= -2.030000
    Process returned 0 (0x0) execution time : 20.680 s
    Press any key to continue.
}
else
}
}
```

Answer to the question no 2

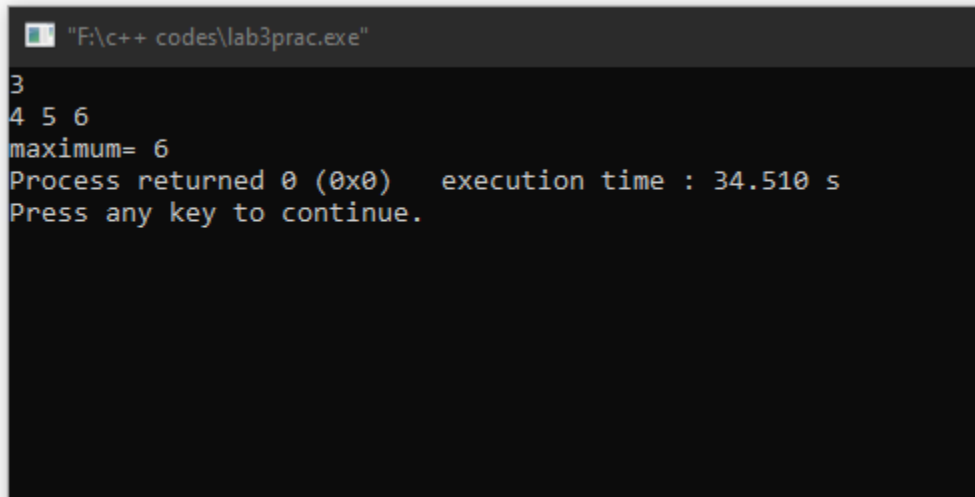
```
#include<stdio.h>

int max_element(int arr[],int n)
{
    int i, max;
    max= arr[0];
    for(i=1;i<n;i++)
        if(arr[i]>max)
            max= arr[i];
    return max;
}

int main (){

    int arr[20],i,n, maximum;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    maximum=max_element(arr,n);
    printf("maximum= %d",maximum);
}
```

Output:



```
"F:\c++ codes\lab3prac.exe"
3
4 5 6
maximum= 6
Process returned 0 (0x0)   execution time : 34.510 s
Press any key to continue.
```

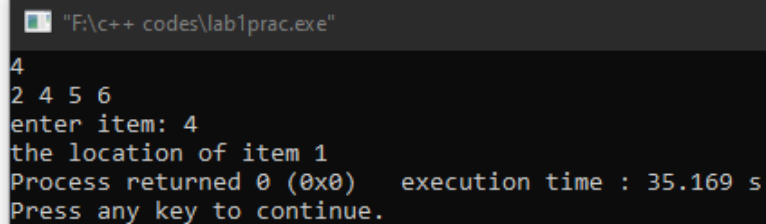
Answer to the question no 3

```
#include<stdio.h>
int search_element(int array[],int n,int m)
{
    int k=1;
    int location=0;
    for(k=1; k<=n; k++){
        if(array[k]==m)
            location=k;
    }
    return location;
}
int main()
{
    int i,n,loc,item,arr[10];

    scanf("%d",&n);

    for(i=0; i<n; i++){
        scanf("%d",&arr[i]);
    }
    printf("enter item: ");
    scanf("%d",&item);
    loc= search_element(arr,n,item);
    if(loc==0)
    {
        printf("item is not in the array list\n");
    }
    else{printf("the location of item %d ",loc);
    }
    return 0;
}
```

Output



```
"F:\c++ codes\lab1prac.exe"
4
2 4 5 6
enter item: 4
the location of item 1
Process returned 0 (0x0)   execution time : 35.169 s
Press any key to continue.
```



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence

COURSE TITLE: data structure

COURSE CODE: cse 232

Lab report 3

Submitted to:

Name: Most. Jannatul Ferdous

Assistant professor

Department of cse

Bangladesh university of business and technology (BUBT)

Submitted by:

Name: Sadia Tasnim

Id : 20211103012

Intake: 46 Section: 1 (day)

Bangladesh university of business and technology (BUB)

Answer to the question no 1

```
#include<stdio.h>
insert_array(int IA[50],int n,int k,int item)
{
    int j=n,i;
    while (j>=k)
    {
        IA[j+1]=IA[j];
        j=j-1;
    }

    IA[k]=item;
    n=n+1;

    printf("element with insert new item : ");

    for (i =0 ; i < n; i++)
        printf("%d",IA[i]);
}

int main()
{
    int a[100], m, i, n, v;
    printf("Enter the elements _ ");
    scanf("%d", &n);

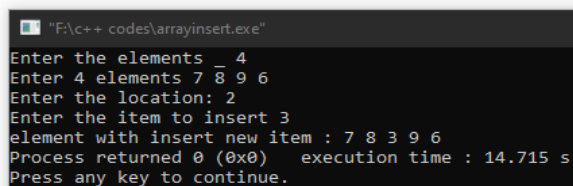
    printf("Enter %d elements ", n);

    for (i =0; i< n; i++)
        scanf("%d", &a[i]);

    printf("Enter the location: ");
    scanf("%d", &m);

    printf("Enter the item to insert ");
    scanf("%d", &i);
    int x=insert_array(a,n,m,i);
}
```

Output:



```
"F:\c++ codes\arrayinsert.exe"
Enter the elements _ 4
Enter 4 elements 7 8 9 6
Enter the location: 2
Enter the item to insert 3
element with insert new item : 7 8 3 9 6
Process returned 0 (0x0)   execution time : 14.715 s
Press any key to continue.
```

Answer to the question no 2

```
#include<stdio.h>
insert_delete(int IA[50],int n,int k)
{
    int i;
    int item;
    item=IA[k];
    int j=k;
    while (j<=n-1)
    {
        IA[j]=IA[j+1];
        j=j+1;
    }
    n=n-1;

    printf("%d item is deleted \n",item);

    for (i = 0; i <n; i++)
        printf("%d ",IA[i]);
}

int main()
{
    int a[50],m,i, n;
    printf("Enter elements number : \n");
    scanf("%d", &n);

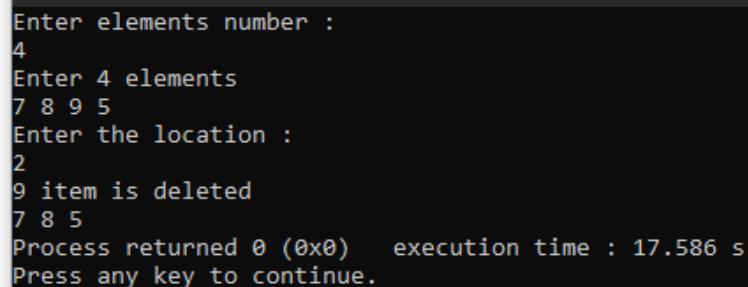
    printf("Enter %d elements\n", n);

    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);

    printf("Enter the location : \n");
    scanf("%d", &m);

    int x=insert_delete(a,n,m);
}
```

Output:



```
Enter elements number :
4
Enter 4 elements
7 8 9 5
Enter the location :
2
9 item is deleted
7 8 5
Process returned 0 (0x0)   execution time : 17.586 s
Press any key to continue.
```



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence

COURSE TITLE: data structure

COURSE CODE: cse 232

Lab report 4

Submitted to:

Name: Most. Jannatul Ferdous

Assistant professor

Department of cse

Bangladesh university of business and technology (BUBT)

Submitted by:

Name: Sadia Tasnim

Id : 20211103012

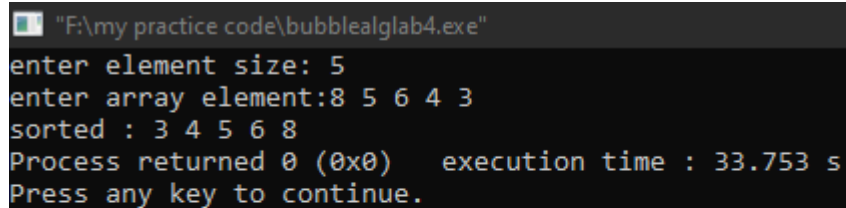
Intake: 46 Section: 1 (day)

Bangladesh university of business and technology (BUB)

Answer to the question no 1

```
#include<stdio.h>
void bubble_sort(int data[], int n)
{
    int i, j, temp;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n-i-1;j++)
        {
            if(data[j] > data[j+1])
            {
                temp = data[j];
                data[j]= data[j+1];
                data[j+1]=temp;
            }
        }
    }
}
int main()
{
    int a,b,N,DATA[10];
    printf("enter element size: ");
    scanf("%d", &N);
    printf("enter array element:");
    for(a=0; a<N; a++)
        scanf("%d",&DATA[a]);
    bubble_sort(DATA,N);
    printf("sorted : ");
    for(a=0;a<N;a++)
        printf("%d ",DATA[a]);
}
```

Output:



```
"F:\my practice code\bubblealglab4.exe"
enter element size: 5
enter array element:8 5 6 4 3
sorted : 3 4 5 6 8
Process returned 0 (0x0)   execution time : 33.753 s
Press any key to continue.
```


Answer to the question no 2

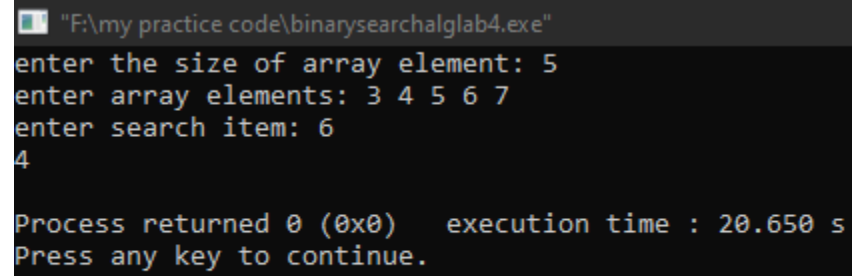
```
#include<stdio.h>
int binary_s(int data[],int END, int item)
{
    int BEG=1,mid;
    mid=((BEG+END)/2);
    while(BEG<=END && data[mid]!=item)
    {

        if (item<data[mid])
        {
            END = mid-1;
        }

        else if(item>data[mid])
        {
            BEG=mid+1;
        }

        mid=((BEG+END)/2);
    }
    return mid;
}
int main()
{
    int DATA[50],bgn,ed,MID,item,LOC,N,binary;
    printf("enter the size of array element: ");
    scanf("%d",&N);
    printf("enter array elements: ");
    for(int i=1; i<=N; i++)
        scanf("%d",&DATA[i]);
    printf("enter search item: ");
    scanf("%d",&item);
    binary = binary_s(DATA,N,item);
    if(DATA[binary]==item){
        LOC = binary;
        printf("%d\n", LOC);
    }
    else
        printf("Item is not found\n");
}
```

Output:



```
"F:\my practice code\binarysearchlab4.exe"  
enter the size of array element: 5  
enter array elements: 3 4 5 6 7  
enter search item: 6  
4  
  
Process returned 0 (0x0)   execution time : 20.650 s  
Press any key to continue.
```



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence

COURSE TITLE: data structure

COURSE CODE: cse 232

Lab report 5

Submitted to:

Name: Most. Jannatul Ferdous

Assistant professor

Department of cse

Bangladesh university of business and technology (BUBT)

Submitted by:

Name: Sadia Tasnim

Id : 20211103012

Intake: 46 Section: 1 (day)

Bangladesh university of business and technology (BUB)

Answer to the question no 1

```
#include<stdio.h>
void insertionsort(int arr[],int n)
{
    for(int i=1;i<n;i++)
    {
        int temp = arr[i];
        int j = i-1;
        while((temp<arr[j])&&(j>=0))
        {
            arr[j+1]=arr[j];
            j=j-1;
        }
        arr[j+1]=temp;
    }

    printf("Sorted elements : ");
    for(int i=0;i<n;i++)
        printf(" %d",arr[i]);
}

int main(){

    int i, j, n, temp, arr[25];

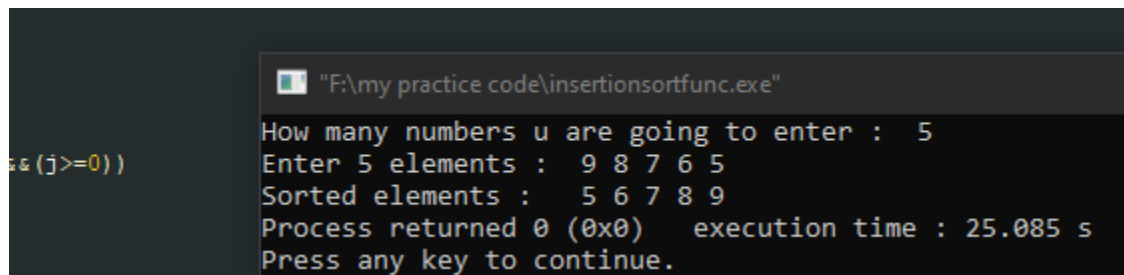
    printf("How many numbers u are going to enter : ");
    scanf("%d",&n);

    printf("Enter %d elements : ", n);

    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    insertionsort(arr,n);

    return 0;
}
```

Output:



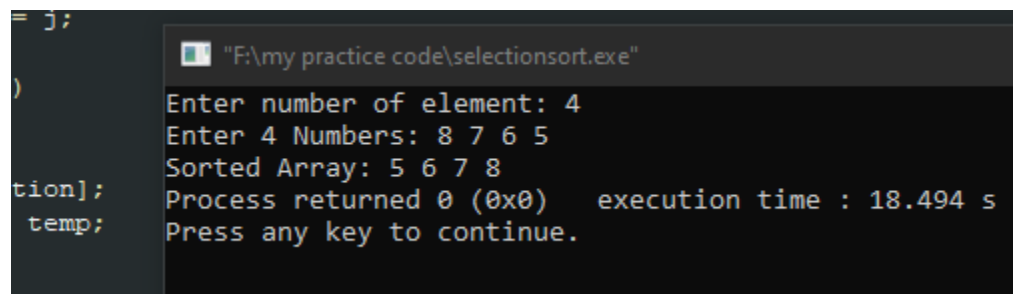
```
"F:\my practice code\insertionsortfunc.exe"
How many numbers u are going to enter : 5
Enter 5 elements : 9 8 7 6 5
Sorted elements : 5 6 7 8 9
Process returned 0 (0x0)   execution time : 25.085 s
Press any key to continue.
```

Answer to the question no 2

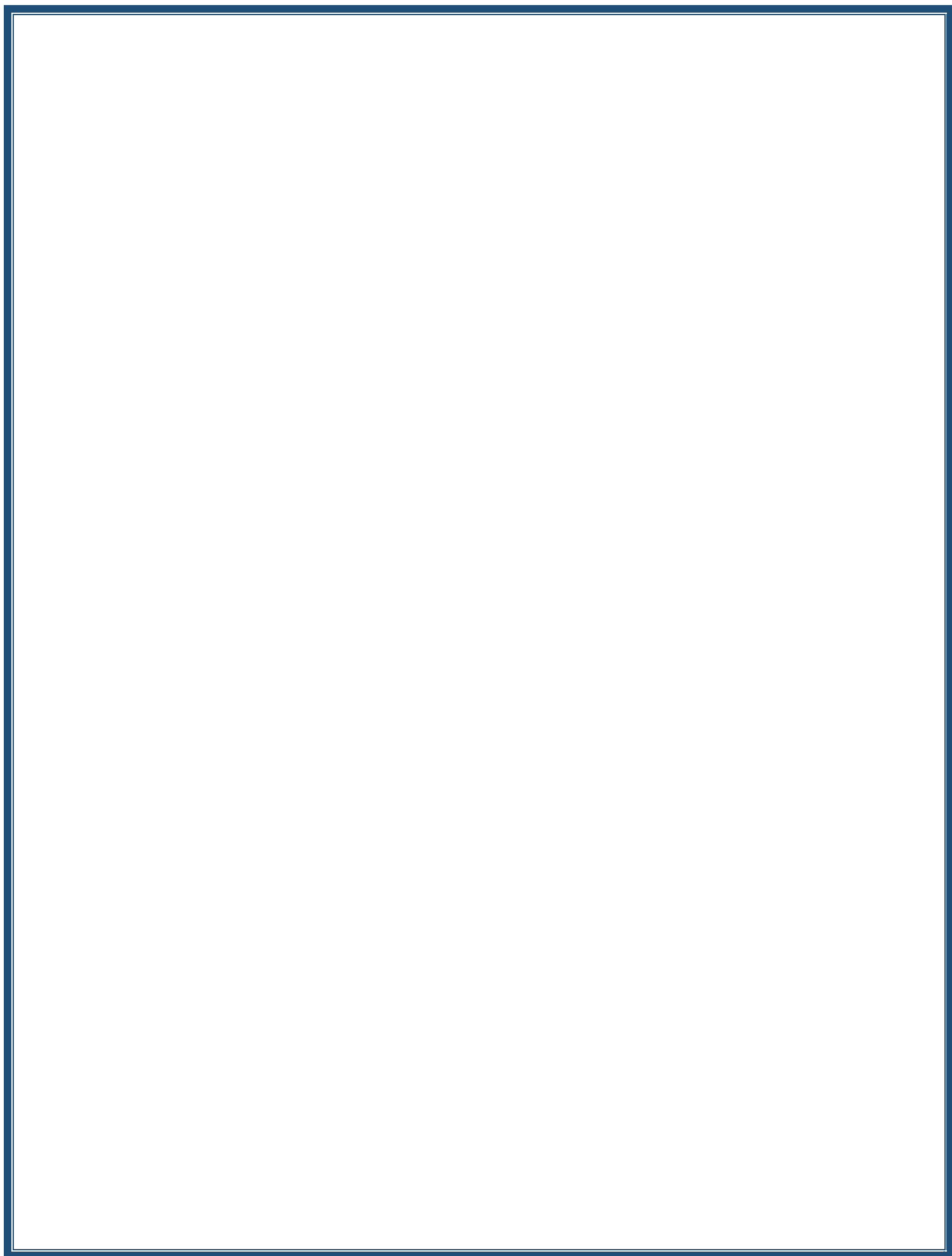
```
#include <stdio.h>
void selectionSor(int a[], int n)
{
    int i, j, position, temp;
    for (i = 0; i < n - 1; i++)
    {
        position = i;
        for (j = i + 1; j < n; j++)
        {
            if (a[position] > a[j])
                position = j;
        }
        if (position != i)
        {
            temp = a[i];
            a[i] = a[position];
            a[position] = temp;
        }
    }
    printf("Sorted Array: ");
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
}
int main()
{
    int a[100], n, i, j, position, temp;
    printf("Enter number of element: ");
    scanf("%d", &n);
    printf("Enter %d Numbers: ", n);
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    selectionSor(a, n);

    return 0;
}
```

Output:



```
= j;
)
tion];
temp;
"F:\my practice code\selectionsort.exe"
Enter number of element: 4
Enter 4 Numbers: 8 7 6 5
Sorted Array: 5 6 7 8
Process returned 0 (0x0)   execution time : 18.494 s
Press any key to continue.
```





BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence

[LAB REPORT 6]

[course code: cse 232]

[Course title: Data structure]

Submitted to:

Most. Jannatul Ferdous

Department of cse (BUBT)

Submitted by:

Name: Sadia Tasnim

ID: 20211103012

Intake: 46 Sec:1

Answer to the question no 1

```
#include<stdio.h>

int stack[50];
int choice,maxstack,top=0,item,i;
void push(int maxstack)
{

    if(top==maxstack)
    {
        printf("\n\tSTACK is overflow");

    }
    else
    {
        printf(" \nEnter a value you want to push : ");
        scanf("%d",&item);
        top++;
        stack[top]=item;
    }

}

void pop()
{

    if(top<=-1)
    {
        printf("\n\t Stack is under flow");
    }

}
```



```
    else
    {
        printf("\n\t The popped elements is %d",stack[top]);
        top--;
    }
}
```

```
void display()
{

    if(top>=0)
    {
        printf("\n The elements in STACK \n");
        for(i=top; i>=0; i--)
            printf("\n%d",stack[i]);
    }
    else
    {
        printf("\n The STACK is empty");
    }

}
```

```
int main()
{

    top=-1;
    printf("\n Enter the size of STACK maximum 50 : ");
    scanf("%d",&maxstack);
```

```
printf("\n\t STACK IMPLEMENTATION");
printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t 4.EXIT");
while(choice!=4)
{
    printf("\n\n Enter the Choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
        {
            push(maxstack);
            break;
        }
        case 2:
        {
            pop();
            break;
        }
        case 3:
        {
            display();
            break;
        }
        case 4:
        {
            printf("\n\t EXITED ");
            break;
        }
    }
```

```
}
```

```
}
```

```
void push();
```

```
void pop();
```

```
void display();
```

```
}
```

Output:

```
Enter the size of STACK maximum 50 : 2

      STACK IMPLEMENTATION
      1.PUSH
      2.POP
      3.DISPLAY
      4.EXIT

Enter the Choice : 1
Enter a value you want to push : 10

Enter the Choice : 1
Enter a value you want to push : 20

Enter the Choice : 3
The elements in STACK
20
10

Enter the Choice : 1
Enter a value you want to push : 30

Enter the Choice : 1
      STACK is overflow

Enter the Choice : 2
      The popped elements is 30

Enter the Choice : 3
The elements in STACK
20
10

Enter the Choice : 4
      EXITED
Process returned 0 (0x0)   execution time : 289.555 s
Press any key to continue.
```



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence



LAB REPORT 7

DATA STRUCTURE LAB



Submitted by:

Name: Most. Jannatul Ferdous

Department of CSE

Submitted by:

Name: Sadia Tasnim

Id : 20211103012

Intake: 46 Section: 1 (day)

ANSWER TO THE QUESTION NO 1

```
#include<bits/stdc++.h>

using namespace std;

int Queues[50],n=3,Front=0,rear=0;

void queueinsert(int item){
    if((Front==1 && rear==n) || Front==rear+1)
        cout<<"\nOVERFLOW....!!\n"<<endl<<endl;
    else if(Front==0){
        Front=1,rear=1;
        Queues[rear]=item;
    }
    else if(rear==n){
        rear=1;
        Queues[rear]=item;
    }
    else{
        rear++;
        Queues[rear]=item;
    }
}

void queuedeleted(){
    int item;
    item = Queues[Front];
    if(Front==0){
        printf("\nUNDERFLOW....!!\n");
    }
}
```

```

    }
    else if(Front==rear){
        printf("\nDELETED DATA: %d\n", item);
        Queues[Front]=NULL;
        Front=0;
        rear=0;
    }
    else if(Front==n){
        printf("\nDELETED DATA: %d\n", item);
        Queues[Front]=NULL;
        Front=1;
    }
    else{
        printf("\nDELETED DATA: %d\n", item);
        Queues[Front]=NULL;
        Front++;
    }
}

```

```

void queuedisplay(){
    int i,sum;

    if(Front>0){
        printf("\n\nALL DATA IN QUEUE: ");
        for(i=1; i<=n; i++){
            cout<<Queues[i]<<" ";
        }
    }
    else

```

```

        cout<<"\nNULL";
    cout<<"\n\n";
}
int main(){
    int C,item,n;
    cout<<"\n....QUEUE IMPLEMENT...\n\n";
    while(C!=4)
    {
        cout<<"\n1.QUEUE INSERT";
        cout<<"\n2.QUEUE DELETE";
        cout<<"\n3.DISPLAY";
        cout<<"\n4.EXIT\n\n";
        cout<<"CHOOSE OPTION: ";
        cin>>C;
        switch(C){
        case 1:
            printf("\n\nENTER DATA ");
            cin>>item;
            queueinsert(item);
            break;

        case 2:
            queuedeleted();
            break;

        case 3:
            queuedisplay();
            break;

        case 4:
            cout<<"EXIT";

```



```
        break;
    }
}

}
```

OUTPUT:

```
....QUEUE IMPLEMENT...
```

```
1.QUEUE INSERT
2.QUEUE DELETE
3.DISPLAY
4.EXIT
```

```
CHOOSE OPTION: 1
```

```
ENTER DATA 1
```

```
1.QUEUE INSERT
2.QUEUE DELETE
3.DISPLAY
4.EXIT
```

```
CHOOSE OPTION: 1
```

```
ENTER DATA 2
```

```
1.QUEUE INSERT
2.QUEUE DELETE
3.DISPLAY
4.EXIT
```

```
CHOOSE OPTION: 1
```

```
ENTER DATA 3
```

```
1.QUEUE INSERT
2.QUEUE DELETE
3.DISPLAY
4.EXIT
```

```
CHOOSE OPTION: 1
```

```
ENTER DATA 4
```

```
OVERFLOW....!!
```

```
1.QUEUE INSERT
2.QUEUE DELETE
3.DISPLAY
4.EXIT
```

3.DISPLAY

4.EXIT

CHOOSE OPTION: 3

ALL DATA IN QUEUE: 1 2 3

1.QUEUE INSERT

2.QUEUE DELETE

3.DISPLAY

4.EXIT

CHOOSE OPTION: 2

DELETED DATA: 1

1.QUEUE INSERT

2.QUEUE DELETE

3.DISPLAY

4.EXIT

CHOOSE OPTION: 2

DELETED DATA: 2

1.QUEUE INSERT

2.QUEUE DELETE

3.DISPLAY

4.EXIT

CHOOSE OPTION: 3

ALL DATA IN QUEUE: 0 0 3

1.QUEUE INSERT

2.QUEUE DELETE

3.DISPLAY

4.EXIT

CHOOSE OPTION: 1

ENTER DATA 4

1.QUEUE INSERT

2.QUEUE DELETE

CHOOSE OPTION: 3

ALL DATA IN QUEUE: 0 0 3

- 1.QUEUE INSERT
- 2.QUEUE DELETE
- 3.DISPLAY
- 4.EXIT

CHOOSE OPTION: 1

ENTER DATA 4

- 1.QUEUE INSERT
- 2.QUEUE DELETE
- 3.DISPLAY
- 4.EXIT

CHOOSE OPTION: 3

ALL DATA IN QUEUE: 4 0 3

- 1.QUEUE INSERT
- 2.QUEUE DELETE
- 3.DISPLAY
- 4.EXIT

CHOOSE OPTION: 4

EXIT

Process returned 0 (0x0) execution time : 162.2

Press any key to continue.



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence



LAB REPORT 8

DATA STRUCTURE LAB



Submitted by:

Name: Most. Jannatul Ferdous

Department of CSE

Submitted by:

Name: Sadia Tasnim

Id : 20211103012

Intake: 46 Section: 1 (day)

ANSWER TO THE QUESTION NO 1

```
#include<bits/stdc++.h>
#include<stdlib.h>
#include<string.h>
using namespace std;

struct node{
    int num;
    node *next;
}*startnode=NULL;

void createnode(int n)
{
    node *tmp,*newnode;

    for(int i =0; i<n; i++)
    {

        newnode=(node *)malloc(sizeof(node));
        cout << "\n\nENTER DATA: ";

        cin>>newnode->num;
        newnode->next=NULL;

        if(startnode==NULL){
```

```

        startnode=newnode;
        tmp=startnode;
    }
    else{
        tmp->next=newnode;
        tmp=tmp->next;
    }

}

}

void display(){

node *tmp;
if(startnode== NULL)
{
    cout<< "\nLIST IS EMPTY..!";
}
else {

    tmp =startnode;
    while(tmp!=NULL)
    {
        printf("\n\n DATA = %x    %d    %x \n",tmp,tmp->num,tmp->next);
        tmp= tmp->next;
    }
}

```

```

}

int main()
{
    int n,x;
    cout<<"\n ENTER NUMBER OF NODES: ";
    cin>>n;
    createnode(n);
    display();

}

```

Output:

 F:\f1\lab8linklistimple.exe

```

ENTER NUMBER OF NODES:  3

ENTER DATA: 1

ENTER DATA: 2

ENTER DATA: 3

DATA = ae1740      1      ae1760

DATA = ae1760      2      ae18c0

DATA = ae18c0      3      0

Process returned 0 (0x0)   execution time : 6.479 s
Press any key to continue.

```

ANSWER TO THE QUESTION NO 2

```
#include<bits/stdc++.h>

#include<stdlib.h>

using namespace std;

struct node{
    int num;
    node *next;
}*startnode=NULL;

void createnode(int n)
{

    node *temp, *newnode;

    for(int i=1; i<=n; i++)
    {
        newnode =(node*)malloc(sizeof(node));
        cout<<"\n\nENTER DATA: ";
        cin>>newnode->num;
        newnode->next=NULL;
        if(startnode==NULL)
        {
            startnode=newnode;
            temp=startnode;
```



```

    }
    else{
        temp->next=newnode;
        temp=temp->next;
    }

}

}

void insert1ST(int data)
{
    node *temp, *newnode;
    newnode=(struct node*)malloc(sizeof(struct node));
    if (newnode==NULL)
    {
        cout<<" MEMORY CAN'T BE ALLOCATED";
    }
    else
    {
        newnode->num=data;
        newnode->next=NULL;
    }
    if(startnode==NULL)
    {
        startnode=newnode;
        cout<<"\n\nNODE INSERTED SUCCESSFULLY AT 1ST";
    }
}

```

```
    }  
    else  
    {  
        temp=startnode;  
        startnode=newnode;  
        newnode->next=temp;  
    }  
}
```

```
void insertlast(int data)
```

```
{  
  
    node *newnode, *a=startnode;  
    newnode=(struct node*)malloc(sizeof(struct node));  
    if(newnode==NULL)  
    {  
  
        cout<<" MEMORY CAN'T BE ALLOCATED";  
    }  
    else{  
        newnode->num=data;  
        newnode->next=NULL;  
    }  
    if(startnode==NULL)  
    {  
        startnode=newnode;  
        cout<<"\n\nNODE INSERTED SUCCESSFULLY AT LAST";  
    }  
}
```

```

    }
    else
    {
        while(a->next!=NULL)
            a=a->next;
        a->next=newnode;
    }
}

```

```

void display()
{
    node *tmp;
    if(startnode==NULL)
    {
        cout<<" LIST IS EMPTY...\n\n";

    }
    else{

        tmp = startnode;
        while(tmp != NULL)
        {
            printf("\n\n DATA = %x   %d   %x \n",tmp,tmp->num,tmp->next);
            tmp = tmp->next;
        }
    }
}

```

```

int main()
{
    int n,x;
    int c,data,position;
    cout<<"\n\tt...IMPLEMENT IN LINK LIST....\n\n";
    cout<<"\n\nDATA ENTERED ON LIST : \n";
    display();
    while(c!=5)
    {
        cout<<"\n\n 1. CREATE\n";
        cout<<" 2. INSERT IN 1ST\n";
        cout<<" 3. INSERT IN LAST\n";
        cout<<" 4. DISPLAY\n";
        cout<<" 5. EXIT\n";
        cout<<"CHOOSE OPTION : ";
        cin>>c;
        switch(c)
        {
            case 1:
                {
                    cout<<"\n ENTER NUMBER OF NODES: ";
                    cin>>n;
                    createnode(n);
                    break;
                }

```

case 2:

```
{  
    cout<<"\n ENTER DATA: ";  
    cin>>data;  
    insert1ST(data);  
    break;  
}
```

case 3:

```
{  
  
    cout<<"\n ENTER DATA: ";  
    cin>>data;  
    insertlast(data);  
    break;  
}
```

case 4:

```
{  
    display();  
    break;  
}
```

case 5:

```
{  
    cout<<"EXIT";  
    break;  
}
```

```

    }

}

return 0;

}

```

OUTPUT:

 "F:\my practice code\LAB8.exe"

```

...IMPLEMENT IN LINK LIST...

DATA ENTERED ON LIST :
LIST IS EMPTY...!

1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. DISPLAY
5. EXIT
CHOOSE OPTION : 1

ENTER NUMBER OF NODES: 2

ENTER DATA: 4

ENTER DATA: 8

1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. DISPLAY
5. EXIT
CHOOSE OPTION : 4

DATA = 751740 4 751760

DATA = 751760 8 0

1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. DISPLAY
5. EXIT
CHOOSE OPTION : 2

ENTER DATA: 2

```

```
1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. DISPLAY
5. EXIT
CHOOSE OPTION : 4
```

```
DATA = 7518c0    2    751740
```

```
DATA = 751740    4    751760
```

```
DATA = 751760    8    0
```

```
1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. DISPLAY
5. EXIT
CHOOSE OPTION : 3
```

```
ENTER DATA: 12
```

```
1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. DISPLAY
5. EXIT
CHOOSE OPTION : 4
```

```
DATA = 7518c0    2    751740
```

```
DATA = 751740    4    751760
```

```
DATA = 751760    8    7518e0
```

```
DATA = 7518e0    12    0
```

```
1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. DISPLAY
5. EXIT
CHOOSE OPTION : 5
EXIT
Process returned 0 (0x0)    execution time : 64.039 s
Press any key to continue.
```



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence



LAB REPORT 8

DATA STRUCTURE LAB



Submitted by:

Name: Most. Jannatul Ferdous

Department of CSE

Submitted by:

Name: Sadia Tasnim

Id : 20211103012

Intake: 46 Section: 1 (day)

ANSWER TO THE QUESTION NO 1

```
#include<bits/stdc++.h>

#include<stdlib.h>

using namespace std;

struct node{
    int num;
    node *next;
}*startnode=NULL;

void createnode(int n)
{

    node *temp, *newnode;

    for(int i=1; i<=n; i++)
    {
        newnode =(node*)malloc(sizeof(node));
        cout<<"\n\nENTER DATA: ";
        cin>>newnode->num;
        newnode->next=NULL;
        if(startnode==NULL)
        {
            startnode=newnode;
            temp=startnode;
```

```
    }  
    else{  
        temp->next=newnode;  
        temp=temp->next;  
    }
```

```
    }  
}
```

```
void insert1ST(int data)  
{  
    node *temp, *newnode;  
    newnode=(struct node*)malloc(sizeof(struct node));  
    if (newnode==NULL)  
    {  
        cout<<" MEMORY CAN'T BE ALLOCATED";  
    }  
    else  
    {  
        newnode->num=data;  
        newnode->next=NULL;  
    }  
    if(startnode==NULL)  
    {  
        startnode=newnode;  
        cout<<"\n\nNODE INSERTED SUCCESSFULLY AT 1ST";  
    }  
    else
```

```
{  
    temp=startnode;  
    startnode=newnode;  
    newnode->next=temp;  
}  
}
```

```
void insertlast(int data)
```

```
{  
  
    node *newnode, *a=startnode;  
    newnode=(struct node*)malloc(sizeof(struct node));  
    if(newnode==NULL)  
    {  
  
        cout<<" MEMORY CAN'T BE ALLOCATED";  
    }  
    else{  
        newnode->num=data;  
        newnode->next=NULL;  
    }  
    if(startnode==NULL)  
    {  
        startnode=newnode;  
        cout<<"\n\nNODE INSERTED SUCCESSFULLY AT LAST";  
    }  
    else
```

```

{
    while(a->next!=NULL)
        a=a->next;
    a->next=newnode;
}
}

```

```

void insertmid(int position,int data)

```

```

{
    node *newnode,*a=startnode;
    newnode= (struct node*)malloc(sizeof(struct node));
    if(newnode==NULL)
    {

        cout<<" MEMORY CAN'T BE ALLOCATED";
    }
    else{
        newnode->num=data;
        newnode->next=NULL;
    }
    for(int i=2; i<position; i++)
    {
        a=a->next;
        if(a==NULL)
            cout<<"LESS THAN"<<position<< "NODES IN THE LIST...!!!";
    }
    newnode->next=a->next;
}

```

```

a->next=newnode;

cout<<"\n\nNODE INSERTED SUCCESSFULLY";

}


void display()
{
    node *tmp;
    if(startnode==NULL)
    {
        cout<<" LIST IS EMPTY...!\n\n\n";

    }
    else{

        tmp = startnode;
        while(tmp != NULL)
        {
            printf("\n\n DATA = %x    %d    %x \n",tmp,tmp->num,tmp->next);
            tmp = tmp->next;
        }
    }
}


void searchh(int x)
{
    node *temp=startnode;
    while(temp!=NULL)

```

```

{
    if(temp->num==x)
    {
        cout<<"FOUND "<<temp->num;
    }
    temp=temp->next;
}

}

int main()
{
    int n,x;
    int c,data,position;
    cout<<"\n\nDATA ENTERED ON LIST : \n";
    display();
    while(c!=7)
    {
        cout<<"\n\n 1. CREATE\n";
        cout<<" 2. INSERT IN 1ST\n";
        cout<<" 3. INSERT IN LAST\n";
        cout<<" 4. INSERT IN MIDDLE\n";
        cout<<" 5. DISPLAY\n";
        cout<<" 6. SEARCH\n";
        cout<<" 7. EXIT\n";
        cout<<"CHOOSE OPTION : ";

```

```
cin>>c;
switch(c)
{
case 1:
    {
        cout<<"\n ENTER NUMBER OF NODES: ";
        cin>>n;
        createnode(n);
        break;
    }

case 2:
    {
        cout<<"\n ENTER DATA: ";
        cin>>data;
        insert1ST(data);
        break;
    }

case 3:
    {

        cout<<"\n ENTER DATA: ";
        cin>>data;
        insertlast(data);
        break;
    }

case 4:
```

```

    {
        cout<<"\n ENTER DATA: ";
        cin>>data;
        cout<<"\n ENTER POSITION: ";
        cin>>position;
        insertmid(position,data);
        break;
    }

    case 5:
    {
        display();
        break;
    }

    case 6:
        cout<<"\nENTER DATA : ";
        cin>>x;
        searchh(x);
        break;

    case 7:
    {
        cout<<"EXIT";
        break;
    }

    }

}

return 0; }

```


Output:

```
DATA ENTERED ON LIST :  
LIST IS EMPTY...!
```

1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. INSERT IN MIDDLE
5. DISPLAY
6. SEARCH
7. EXIT

```
CHOOSE OPTION : 1
```

```
ENTER NUMBER OF NODES: 3
```

```
ENTER DATA: 1
```

```
ENTER DATA: 3
```

```
ENTER DATA: 4
```

1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. INSERT IN MIDDLE
5. DISPLAY
6. SEARCH
7. EXIT

```
CHOOSE OPTION : 5
```

```
DATA = d91740      1      d91760
```

```
DATA = d91760      3      d918c0
```

```
DATA = d918c0      4      0
```

```
1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. INSERT IN MIDDLE
5. DISPLAY
6. SEARCH
7. EXIT
CHOOSE OPTION : 4

ENTER DATA: 2

ENTER POSITION: 2

NODE INSERTED SUCCESSFULLY

1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. INSERT IN MIDDLE
5. DISPLAY
6. SEARCH
7. EXIT
CHOOSE OPTION : 5

DATA = d91740      1      d918e0

DATA = d918e0      2      d91760

DATA = d91760      3      d918c0

DATA = d918c0      4      0

1. CREATE
2. INSERT IN 1ST
3. INSERT IN LAST
4. INSERT IN MIDDLE
5. DISPLAY
6. SEARCH
7. EXIT
CHOOSE OPTION : 7
EXIT
Process returned 0 (0x0)   execution time : 105.851 s
Press any key to continue.
```

ANSWER TO THE QUESTION NO 2

```
#include<bits/stdc++.h>

#include<stdlib.h>

using namespace std;

struct node{
    int num;
    node *next;
}*startnode=NULL;

void createnode(int n)
{
    node *temp, *newnode;

    for(int i=1; i<=n; i++)
    {
        newnode =(node*)malloc(sizeof(node));
        cout<<"\n\nENTER DATA: ";
        cin>>newnode->num;
        newnode->next=NULL;
        if(startnode==NULL)
        {
            startnode=newnode;
            temp=startnode;
        }
    }
}
```

```
else{  
    temp->next=newnode;  
    temp=temp->next;  
}
```

```
}
```

```
}
```

```
void delete1ST(int data)  
{  
    node *newnode;  
    if(startnode==NULL)  
    {  
        cout<<"\n\nERROR.....LIST IS EMPTY....!!!";  
    }  
    else  
    {  
        node* temp=startnode;  
        startnode=temp->next;  
        delete temp;  
        cout<<"\n\nFIRST NODE DELETED SUCCESSFULLY...!!!";  
    }  
}
```

```

void deletelast(int data)
{
    node *newnode;
    if(startnode==NULL)
    {
        cout<<"\n\nERROR.....LIST IS EMPTY....!!!";
    }
    else
    {
        node *q=startnode;
        while(q->next->next!=NULL)
            q=q->next;
        node* temp=q->next;
        q->next=NULL;
        delete temp;
        cout<<"\n\nLAST NODE DELETED SUCCESSFULLY...!!";
    }
}

```

```

void deletemid(int position,int data)
{
    node *newnode,*a=startnode;
    for(int i=2; i<position; i++)
    {
        a=a->next;
        if(a==NULL)
            cout<<"\n\nNOODE NOT FOUND\n";
    }
}

```

```

    }

    if(int i=position)

    newnode=a->next;
    a->next=newnode->next;
    delete newnode;
    cout<<"\n\nNODE DELETED SUCCESSFULLY";

}

```

```

void display()
{
    node *tmp;
    if(startnode==NULL)
    {
        cout<<" LIST IS EMPTY...!\n\n\n";

    }
    else{

        tmp = startnode;
        while(tmp != NULL)
        {
            printf("\n\n DATA = %x   %d   %x \n",tmp,tmp->num,tmp->next);
            tmp = tmp->next;
        }
    }
}

```

```
int main()
{
    int n,x;
    int c,data,position;
    cout<<"\n\nDATA ENTERED ON LIST : \n";
    display();
    while(c!=6)
    {
        cout<<"\n\n 1. CREATE NODE\n";
        cout<<" 2. DELETE 1ST NODE\n";
        cout<<" 3. DELETE LAST NODE\n";
        cout<<" 4. DELETE MIDDLE NODE\n";
        cout<<" 5. DISPLAY\n";
        cout<<" 6.EXIT\n";
        cout<<"CHOOSE OPTION : ";
        cin>>c;
        switch(c)
        {
            case 1:
            {
                cout<<"\n ENTER NUMBER OF NODES: ";
                cin>>n;
                createnode(n);
                break;
            }
        }
    }
}
```

case 2:

```
{  
    delete1ST(data);  
    break;  
}
```

case 3:

```
{  
    deletelast(data);  
    break;  
}
```

case 4:

```
{  
    cout<<"\n ENTER POSITION: ";  
    cin>>position;  
    deletemid(position,data);  
    break;  
}
```

case 5:

```
{  
    display();  
    break;  
}
```

case 6:

```
{  
    cout<<"EXIT";
```



```
        break;
    }

}
}
return 0;

}
```

OUTPUT:

F:\f1\lab9linklistdelete.exe

DATA ENTERED ON LIST :
LIST IS EMPTY...!

1. CREATE NODE
2. DELETE 1ST NODE
3. DELETE LAST NODE
4. DELETE MIDDLE NODE
5. DISPLAY
- 6.EXIT

CHOOSE OPTION : 1

ENTER NUMBER OF NODES: 4

ENTER DATA: 1

ENTER DATA: 2

ENTER DATA: 3

ENTER DATA: 4

1. CREATE NODE
2. DELETE 1ST NODE
3. DELETE LAST NODE
4. DELETE MIDDLE NODE
5. DISPLAY
- 6.EXIT

CHOOSE OPTION : 2

FIRST NODE DELETED SUCCESSFULLY...!!

1. CREATE NODE
2. DELETE 1ST NODE
3. DELETE LAST NODE
4. DELETE MIDDLE NODE
5. DISPLAY
- 6.EXIT

CHOOSE OPTION : 5

```
DATA = 1e1760      2      1e18c0
```

```
DATA = 1e18c0      3      1e18e0
```

```
DATA = 1e18e0      4      0
```

1. CREATE NODE
2. DELETE 1ST NODE
3. DELETE LAST NODE
4. DELETE MIDDLE NODE
5. DISPLAY
- 6.EXIT

```
CHOOSE OPTION : 3
```

```
LAST NODE DELETED SUCCESSFULLY...!!
```

1. CREATE NODE
2. DELETE 1ST NODE
3. DELETE LAST NODE
4. DELETE MIDDLE NODE
5. DISPLAY
- 6.EXIT

```
CHOOSE OPTION : 5
```

```
DATA = 1e1760      2      1e18c0
```

```
DATA = 1e18c0      3      0
```

1. CREATE NODE
2. DELETE 1ST NODE
3. DELETE LAST NODE
4. DELETE MIDDLE NODE
5. DISPLAY
- 6.EXIT

```
CHOOSE OPTION : 4
```

```
ENTER POSITION: 2
```

```
NODE DELETED SUCCESSFULLY
```

ENTER POSITION: 2

NODE DELETED SUCCESSFULLY

1. CREATE NODE
2. DELETE 1ST NODE
3. DELETE LAST NODE
4. DELETE MIDDLE NODE
5. DISPLAY
- 6.EXIT

CHOOSE OPTION : 5

DATA = 1e1760 2 0

1. CREATE NODE
2. DELETE 1ST NODE
3. DELETE LAST NODE
4. DELETE MIDDLE NODE
5. DISPLAY
- 6.EXIT

CHOOSE OPTION : 6

EXIT

Process returned 0 (0x0) execution time : 59.170 s

Press any key to continue.



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence



LAB REPORT 10

DATA STRUCTURE LAB



Submitted to:

Name: Most. Jannatul Ferdous

Department of CSE

Submitted by:

Name: Sadia Tasnim

Id : 20211103012

Intake: 46 Section: 1 (day)

Answer to the problem solution 1

```
#include<bits/stdc++.h>

#include<stdlib.h>

using namespace std;

struct node{
    int num;
    node *next;
}*startnode=NULL;

void createnode(int n)
{

    node *temp, *newnode;

    for(int i=1; i<=n; i++)
    {
        newnode =(node*)malloc(sizeof(node));
        cout<<"\n\nENTER DATA: ";
        cin>>newnode->num;
        newnode->next=NULL;
        if(startnode==NULL)
        {
            startnode=newnode;
            temp=startnode;
```

```
}  
else{  
    temp->next=newnode;  
    temp=temp->next;  
}
```

```
}  
}
```

```
void searchh(int x)  
{  
    node *temp=startnode;  
    while(temp!=NULL)  
    {  
        if(temp->num==x)  
        {  
            cout<<"\n\n\n>>>--DATA FOUND : "<<temp->num;  
            break;  
        }  
        else  
        {  
            temp=temp->next;  
            cout<<"\n\n\n.....NOT FOUND.....\n";  
            break;  
        }  
    }  
}
```

```
}
```

```
void display()
```

```
{
```

```
    node *tmp;
```

```
    if(startnode==NULL)
```

```
    {
```

```
        cout<<" LIST IS EMPTY...!\n\n";
```

```
    }
```

```
    else{
```

```
        tmp = startnode;
```

```
        while(tmp != NULL)
```

```
        {
```

```
            printf("\n\n DATA = %x    %d    %x \n",tmp,tmp->num,tmp->next);
```

```
            tmp = tmp->next;
```

```
        }
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int n,x;
```

```
    int c,data,position;
```



```
cout<<"\n\nDATA ENTERED ON LIST : \n";
while(c!=4)
{
    cout<<"\n\n 1. CREATE\n";
    cout<<" 2. DISPLAY\n";
    cout<<" 3. SEARCH\n";
    cout<<" 4. EXIT\n";
    cout<<"CHOOSE OPTION : ";
    cin>>c;
    switch(c)
    {
    case 1:
        {
            cout<<"\n ENTER NUMBER OF NODES: ";
            cin>>n;
            createnode(n);
            break;
        }

    case 2:
        {
            display();
            break;
        }

    case 3:
        {
            cout<<"\nENTER DATA : ";
            cin>>x;
            searchh(x);
            break;
        }
    }
```

```
case 4:  
{  
    cout<<"EXIT";  
    break;  
}  
  
}  
}  
return 0;  
  
}
```

Output:

```
DATA ENTERED ON LIST :
```

1. CREATE
2. DISPLAY
3. SEARCH
4. EXIT

```
CHOOSE OPTION : 1
```

```
ENTER NUMBER OF NODES: 2
```

```
ENTER DATA: 5
```

```
ENTER DATA: 10
```

1. CREATE
2. DISPLAY
3. SEARCH
4. EXIT

```
CHOOSE OPTION : 2
```

```
DATA = 9d1750      5      9d1770
```

```
DATA = 9d1770      10      0
```

1. CREATE
2. DISPLAY
3. SEARCH
4. EXIT

```
CHOOSE OPTION : 3
```

```
ENTER DATA : 2
```

```
.....NOT FOUND.....
```

```
.....NOT FOUND.....
```

```
1. CREATE  
2. DISPLAY  
3. SEARCH  
4. EXIT
```

```
CHOOSE OPTION : 3
```

```
ENTER DATA : 5
```

```
>>>--DATA FOUND : 5
```

```
1. CREATE  
2. DISPLAY  
3. SEARCH  
4. EXIT
```

```
CHOOSE OPTION : 4
```

```
EXIT
```

```
Process returned 0 (0x0) execution time : 43.835 s
```

```
Press any key to continue.
```

Answer to the problem solution 2

```
#include <iostream>
```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
struct node
```

```
{
```

```
    int num;
```

```
    node *next, *prev;
```

```
} *stnode = NULL, *ennode = NULL;
```

```
void createnode(int n);
```

```
void display();
```

```
int main()
```

```
{
```

```
    int n;
```

```
    cout << "\nEnter No Of Nodes: ";
```

```
    cin >> n;
```

```
    createnode(n);
```

```
    display();
```

```
}
```

```
void createnode(int n)
```

```
{
```

```
    node *tmp, *newn;
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

```
        newn = (node *)malloc(sizeof(node));
```

```
        cout << "\nEnter Data: ";
```

```
        cin >> newn->num;
```

```
        newn->num;
```

```
        newn->next = NULL;
```

```
        newn->prev = ennode;
```

```
        if (stnode == NULL)
```

```
        {
```

```
            stnode = newn;
```

```
            ennode = stnode;
```

```
        }
```

```
    else
```

```
    {
```

```

        ennode->next = newn;
        ennode = ennode->next;
    }
}

void display()
{
    node *tmp;
    if (stnode == NULL)
    {
        cout << "List Is Empty" << endl;
    }
    else
    {
        tmp = stnode;
        while (tmp != NULL)
        {
            cout << "Data = " << tmp->prev << " " << tmp << " " << tmp->num << " " << tmp->next << endl;
            tmp = tmp->next;
        }
    }
}

```

Output:

"F:\All CODES\2nd semester code\lab 10.exe"

Enter No Of Nodes: 4

Enter Data: 1

Enter Data: 2

Enter Data: 3

Enter Data: 4

Data = 0 0x6d1750 1 0x6d1770

Data = 0x6d1750 0x6d1770 2 0x6d18d0

Data = 0x6d1770 0x6d18d0 3 0x6d18f0

Data = 0x6d18d0 0x6d18f0 4 0

Process returned 0 (0x0) execution time : 7.860 s

Press any key to continue.