Aravali College Of Engineering And Management

Practical File On

Data Structure Analysis

File Submitted By :-	**File Submitted To :-**
Name :- Tushar Gera	**Name :- Sudha Ma'am**
Roll No :- 22BCADS54	
Branch :- BCA(Data Science)	

Index

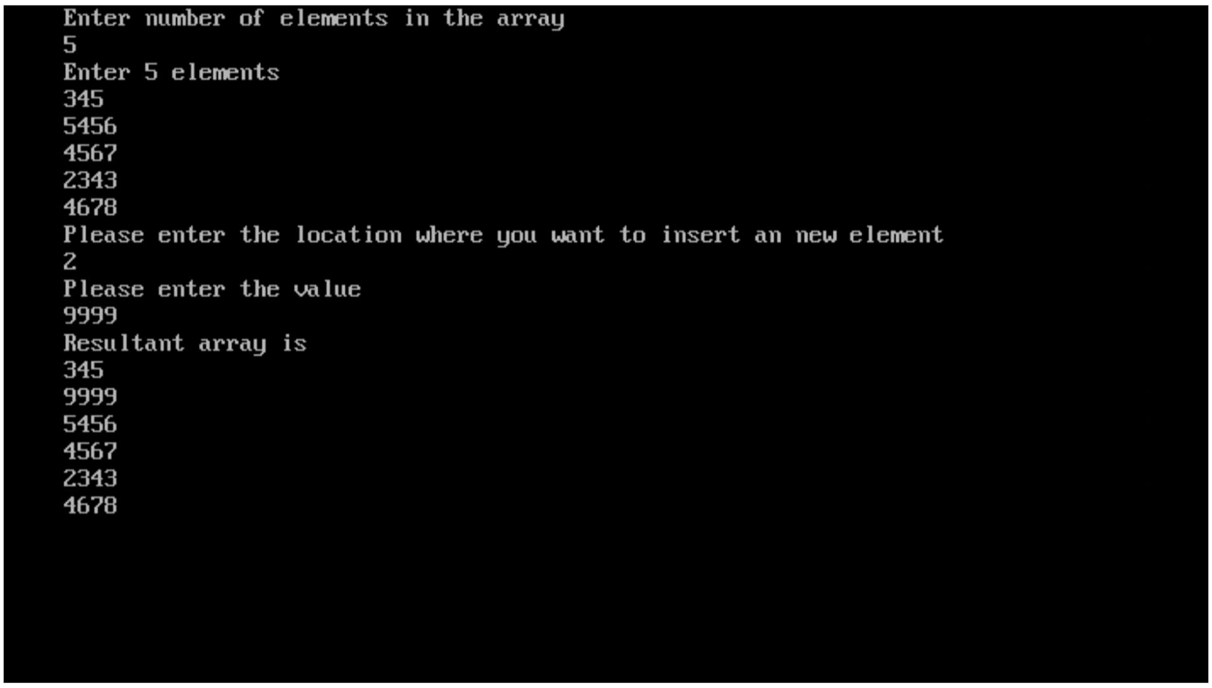
PROGRAM NO 1

AIM : WRITE A PROGRAM TO INSERT AN ELEMENT IN AN ARRAY

CODE :

```
#include<stdio.h>
#include<conio.h>
voidmain()
{
   int array[50], position, c, n, value;

printf("Enter number of elements in the array\n");
scanf("%d", &n);

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

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

printf("Please enter the location where you want to insert an new element\n");
scanf("%d", &position);

printf("Please enter the value\n");
scanf("%d", &value);

   for (c = n - 1; c >= position - 1; c--)
      array[c+1] = array[c];

   array[position-1] = value;

printf("Resultant array is\n");

   for (c = 0; c <= n; c++)
printf("%d\n", array[c]);

getch();
}
```

OUTPUT :



```
Enter number of elements in the array
5
Enter 5 elements
345
5456
4567
2343
4678
Please enter the location where you want to insert an new element
2
Please enter the value
9999
Resultant array is
345
9999
5456
4567
2343
4678
```

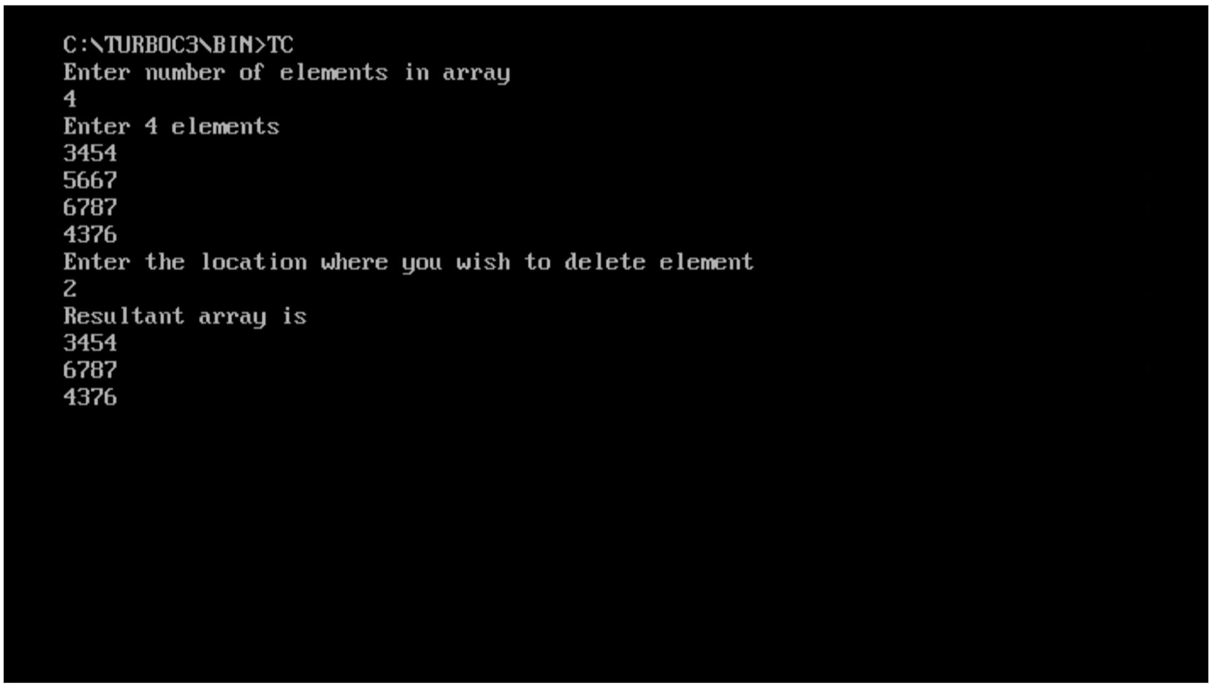
PROGRAM NO 2

AIM : WRITE A PROGRAM TO DELETE AN ELEMENT FROM AN ARRAY

CODE :

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int array[100], position, c, n;

printf("Enter number of elements in array\n");
scanf("%d", &n);

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

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

printf("Enter the location where you wish to delete element\n");
scanf("%d", &position);

   if ( position>= n+1 )
printf("Deletion not possible.\n");

   else
   {
      for ( c = position - 1 ; c < n - 1 ; c++ )
      array[c] = array[c+1];

printf("Resultant array is\n");

for( c = 0 ; c < n - 1 ; c++ )
printf("%d\n", array[c]);
   }
getch();
}
```

OUTPUT :



```
C:\TURBOC3\BIN>TC
Enter number of elements in array
4
Enter 4 elements
3454
5667
6787
4376
Enter the location where you wish to delete element
2
Resultant array is
3454
6787
4376
```

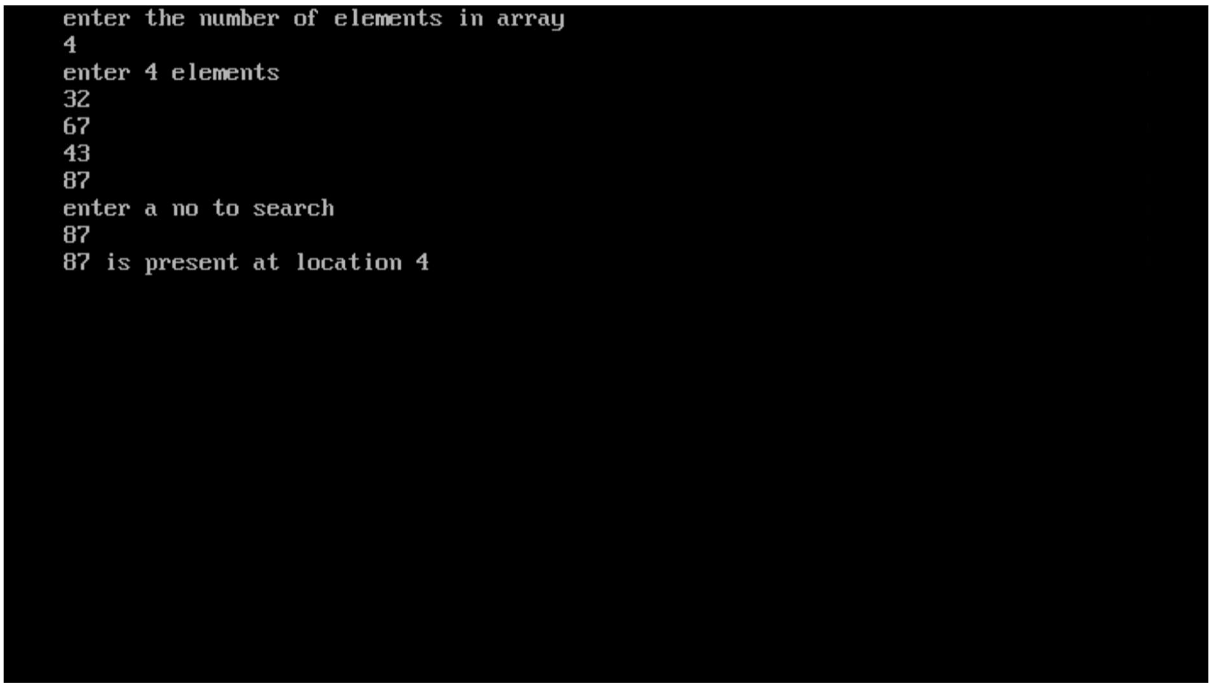
PROGRAM NO 3

AIM :Write a program for searching an element in an array (linear search)

CODE :

```
#include<stdio.h>
#include<conio.h>
void main()
{
int array[100],search,c,n;
clrscr();
printf("enter the number of elements in array\n");
scanf("%d",&n);
printf("enter %d elements \n",n);
for(c=0;c<n;c++)
{
scanf("%d",&array[c]);
}
printf("enter a no to search \n");
scanf("%d",&search);
for(c=0;c<n;c++)
{
if(array[c]==search)
{
printf("%d is present at location %d\n",search,c+1);
break;
}
if(c==n)
{
printf("%d is not present in the array \n",search);
}
}
getch();
}
```

OUTPUT :



```
enter the number of elements in array
4
enter 4 elements
32
67
43
87
enter a no to search
87
87 is present at location 4
```

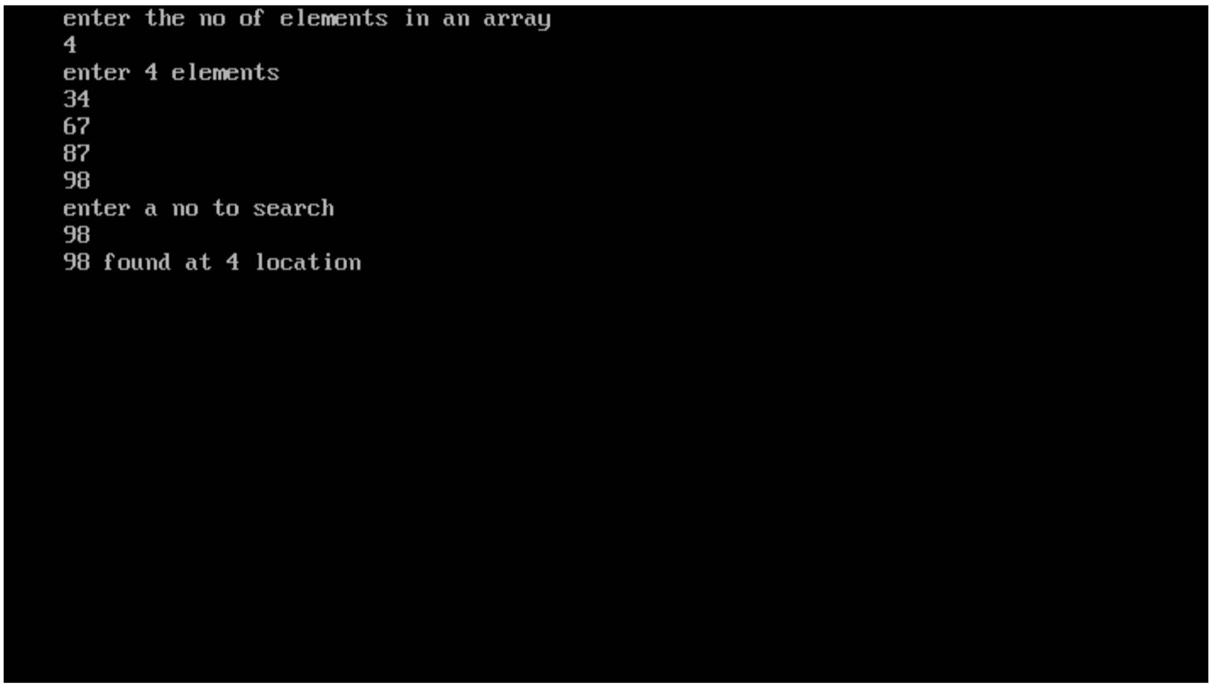
PROGRAM NO 4

AIM : Write a program for searching an element in an array (binary search)

CODE :

```
#include<stdio.h>
#include<conio.h>
void main()
{
int array[100],s,n,last,mid,first,c;
clrscr();
printf("enter the no of elements in an array \n");
scanf("%d",&n);
printf("enter %d elements\n",n);
for(c=0;c<n;c++)
scanf("%d",&array[c]);
printf("enter a no to search \n");
scanf("%d",&s);
first=0;
last=n-1;
mid=(first+last)/2;
while(first<=last)
{
if(array[mid]<s)
first=mid+1;
else if(array[mid]==s)
{
printf("%d found at %d location\n",s,mid+1);
break;
}
else
last=mid-1;
mid=(first+last)/2;
}
if(first>last)
printf("%d not found \n",s);
getch();
}
```

OUTPUT :



```
enter the no of elements in an array
4
enter 4 elements
34
67
87
98
enter a no to search
98
98 found at 4 location
```

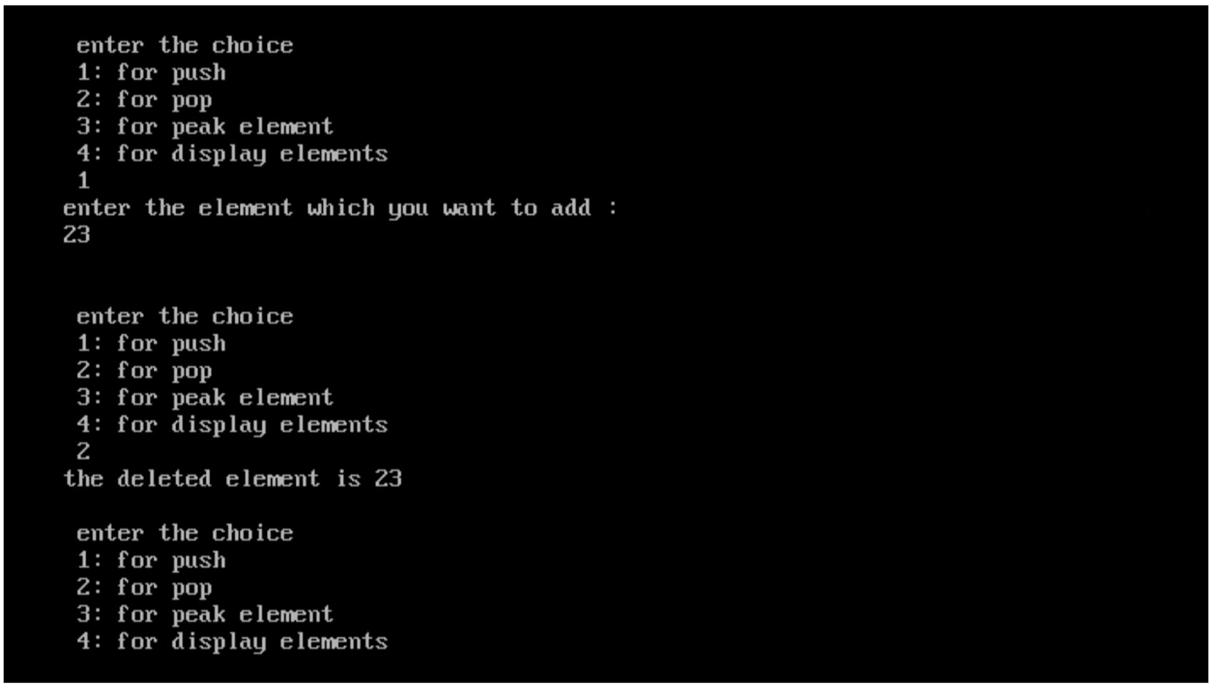
PROGRAM NO 5

AIM :Write a program for push, pop, peak, display.

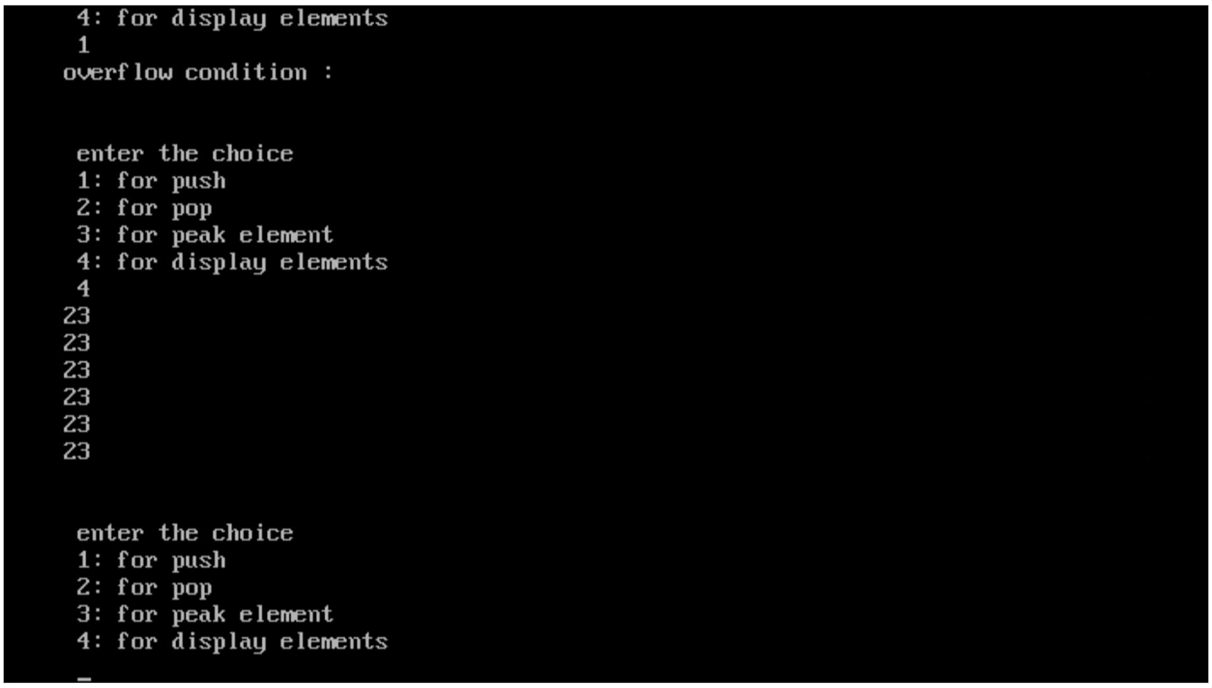
CODE :

```
#include<stdio.h>
#include<conio.h>
#define N 6
void push();
void pop();
void peak();
void display();
int stack[N];
int top=-1;
void main()
{
        int ch;
        clrscr();
                do
                {
                        printf("\n\n enter the choice \n 1: for push \n 2: for pop \n 3:
for peak element \n 4: for display elements \n ");
                        scanf("%d",&ch);
                        switch(ch)
                        {
                                case 1:push();
                                                break;
                                case 2:pop();
                                                break;
                                case 3:peak();
                                                break;
                                case 4:display();
                                                break;
                                case 0:printf("program ends now \n");
                                                break;
                                default :printf("invalid input");
                                                        break;
                        }
                }while(ch!=0);
                getch();
}
void push()
{
        int input;
        if(top==N-1)
        {
                printf("overflow condition :\n");
        }
        else
        {
                printf("enter the element which you want to add :\n");
                scanf("%d",&input);
                top++;
                stack[top]=input;
        }
}
void pop()
{
        int deleted;
        if(top==-1)
        {
                printf("underflow condition :\n");
        }
        else
        {
                deleted=stack[top];
                printf("the deleted element is %d ",deleted);
                top--;
        }
}
void peak()
{
        if(top==-1)
        {
                printf("no element is there in the stack \n");
        }
        else
        {
        printf("the peak element is %d ",stack[top]);
        }
}
void display()
{
        int i;
        for(i=top;i>=0;i--)
        {
                printf("%d\n",stack[i]);
        }
}
```

OUTPUT :



```
  enter the choice
  1: for push
  2: for pop
  3: for peak element
  4: for display elements
  1
 enter the element which you want to add :
 23


  enter the choice
  1: for push
  2: for pop
  3: for peak element
  4: for display elements
  2
 the deleted element is 23

  enter the choice
  1: for push
  2: for pop
  3: for peak element
  4: for display elements
```



```
  4: for display elements
  1
 overflow condition :


  enter the choice
  1: for push
  2: for pop
  3: for peak element
  4: for display elements
  4
 23
 23
 23
 23
 23
 23


  enter the choice
  1: for push
  2: for pop
  3: for peak element
  4: for display elements
  _
```

PROGRAM NO 6

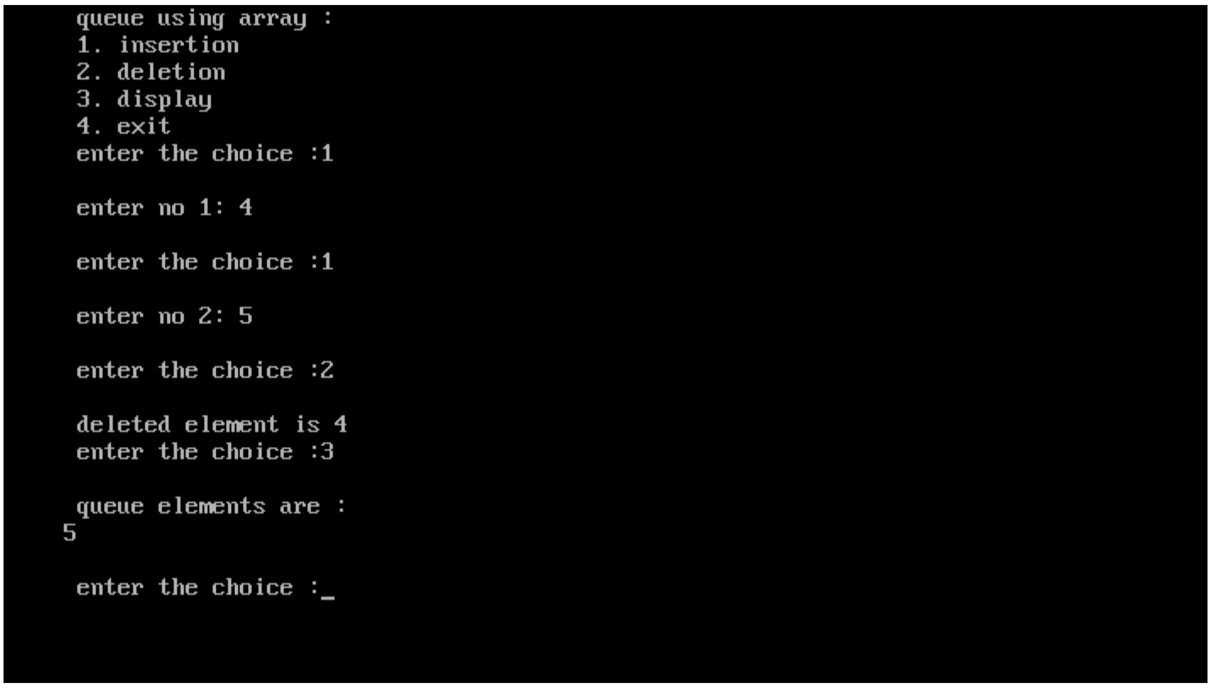
AIM :Write a program for insertion, deletion and display in queue.

CODE :

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define n 5
void main()
{
int queue[n],ch=1,front=0,rear=0,i,j=1,x=n;
clrscr();
printf(" queue using array :");
printf("\n 1. insertion \n2. deletion \n3. display \n4. exit");
while(ch)
{
printf("\n enter the choice :");
scanf("%d",&ch);
switch(ch)
{
case 1:
if(rear==x)
printf(" queue is full \n");
else
{
printf("\n enter no %d: ",j++);
scanf("%d",&queue[rear++]);
}
break;
case 2:
if(front==rear)
{
printf(" queue is empty");
}
else
{
printf("\n deleted element is %d ",queue[front++]);
}
break;
case 3:
```

```
if(front==rear)
{
printf(" queue is empty");
}
else
{
printf("\n queue elements are :\n");
for(i=front;i<rear;i++)
{
printf("%d",queue[i]);
printf("\n");
}
break;
case 4:
exit(0);
default :
printf("\n wrong choice : please see the options ");
}
}
}
getch();
}
```

OUTPUT :



```
     queue using array :
     1. insertion
     2. deletion
     3. display
     4. exit
     enter the choice :1

     enter no 1: 4

     enter the choice :1

     enter no 2: 5

     enter the choice :2

     deleted element is 4
     enter the choice :3

     queue elements are :
    5

     enter the choice :_
```

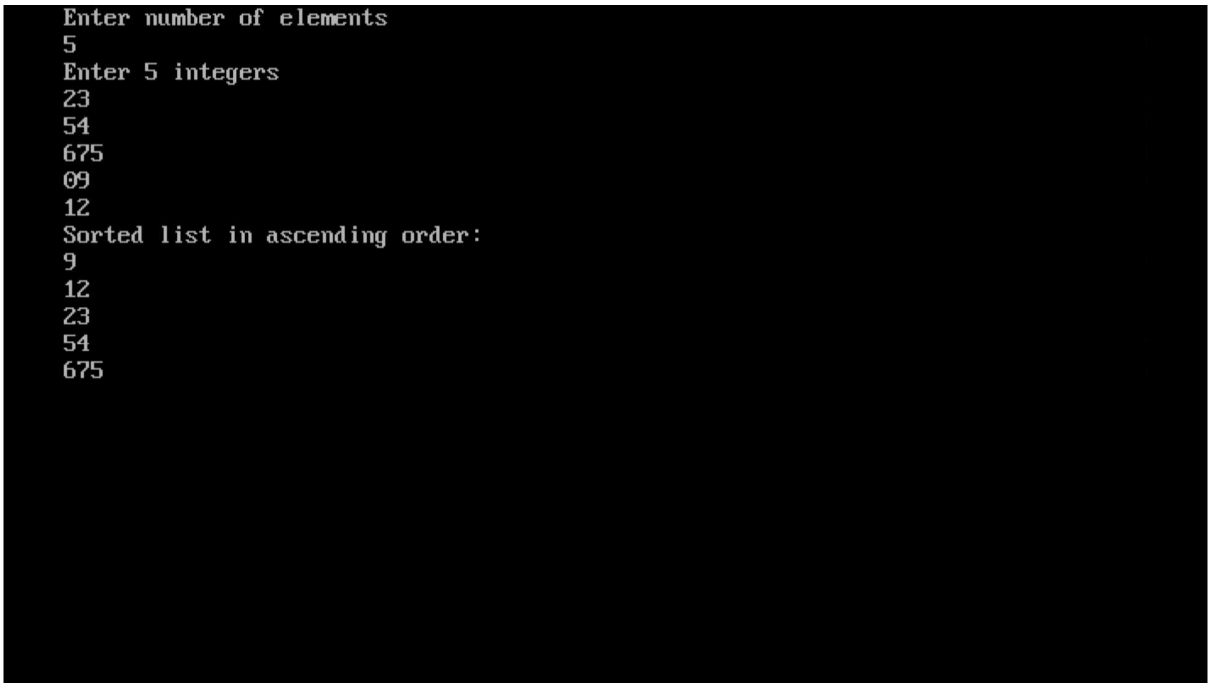
PROGRAM NO 7

AIM :Write a program of Bubble sorting

CODE :

```
#include <stdio.h>

int main()
{
  int array[100], n, c, d, swap;

  printf("Enter number of elements\n");
  scanf("%d", &n);

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

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

  for (c = 0 ; c < n - 1; c++)
  {
    for (d = 0 ; d < n - c - 1; d++)
    {
      if (array[d] > array[d+1]) /* For decreasing order use < */
      {
        swap       = array[d];
        array[d]   = array[d+1];
        array[d+1] = swap;
      }
    }
  }

  printf("Sorted list in ascending order:\n");

  for (c = 0; c < n; c++)
     printf("%d\n", array[c]);

  return 0;
}
```

OUTPUT :



```
Enter number of elements
5
Enter 5 integers
23
54
675
09
12
Sorted list in ascending order:
9
12
23
54
675
```

PROGRAM NO 8

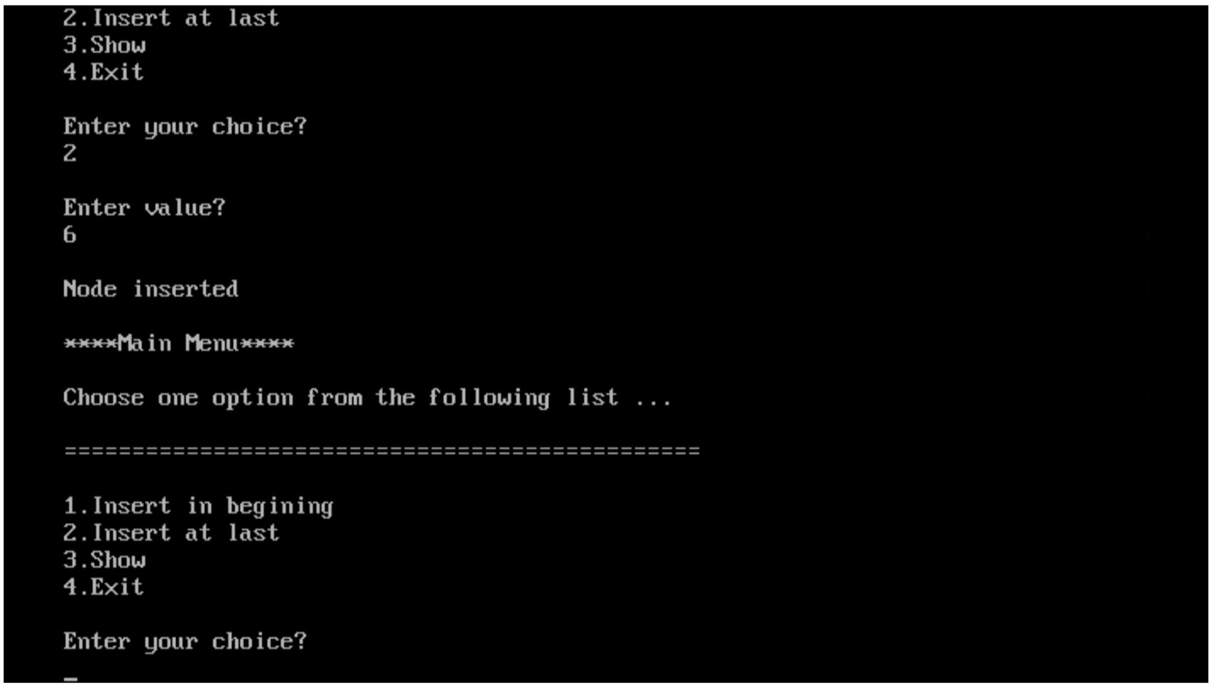
AIM :Write a program of Link list implementation for insertion

CODE :

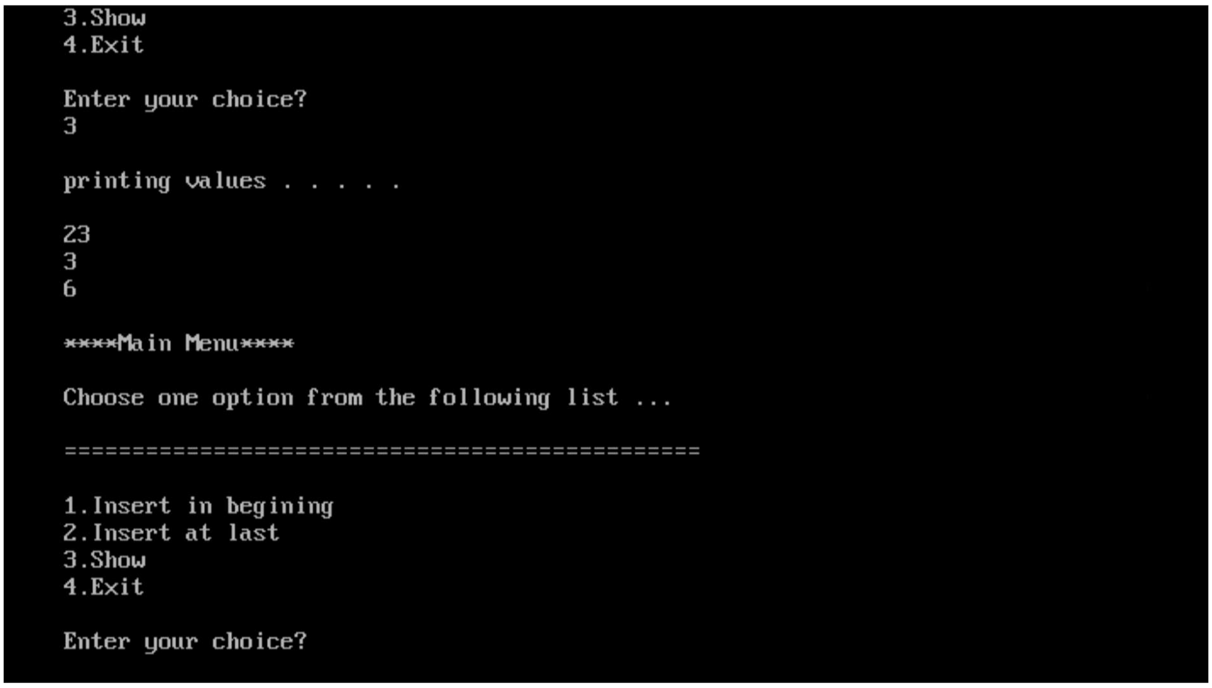
```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int data;
   struct node *next;
};
struct node *head;

void beginsert ();
void lastinsert ();
void display();

void main ()
{
   int choice =0;
   while(choice != 4)
   {
      printf("\n\n****Main Menu****\n");
      printf("\nChoose one option from the following list ...\n");

printf("\n===============================================\n");
      printf("\n1.Insert in begining\n2.Insert at last\n3.Show\n4.Exit\n");
      printf("\nEnter your choice?\n");
      scanf("\n%d",&choice);
      switch(choice)
      {
         case 1:
         beginsert();
         break;
         case 2:
         lastinsert();
         break;


         case 3:
         display();
         break;
```

```
            case 4:
            exit(0);
            break;
            default:
            printf("Please enter valid choice..");
        }
    }
}
void beginsert()
{
    struct node *ptr;
    int item;
    ptr = (struct node *) malloc(sizeof(struct node *));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter value\n");
        scanf("%d",&item);
        ptr->data = item;
        ptr->next = head;
        head = ptr;
        printf("\nNode inserted");
    }

}
void lastinsert()
{
    struct node *ptr,*temp;
    int item;
    ptr = (struct node*)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter value?\n");
        scanf("%d",&item);
        ptr->data = item;
        if(head == NULL)
        {
            ptr -> next = NULL;
            head = ptr;
            printf("\nNode inserted");
        }
        else
        {
            temp = head;
            while (temp -> next != NULL)
            {
               temp = temp -> next;
            }
            temp->next = ptr;
            ptr->next = NULL;
            printf("\nNode inserted");

        }
    }
}

void display()
{
    struct node *ptr;
    ptr = head;
    if(ptr == NULL)
    {
        printf("Nothing to print");
    }
    else
    {
        printf("\nprinting values . . . . .\n");
        while (ptr!=NULL)
        {
            printf("\n%d",ptr->data);
            ptr = ptr -> next;
        }
    }
}
```

OUTPUT :



```
2.Insert at last
3.Show
4.Exit

Enter your choice?
2

Enter value?
6

Node inserted

****Main Menu****

Choose one option from the following list ...

===============================================

1.Insert in begining
2.Insert at last
3.Show
4.Exit

Enter your choice?
_
```



```
3.Show
4.Exit

Enter your choice?
3

printing values . . . . .

23
3
6

****Main Menu****

Choose one option from the following list ...

===============================================

1.Insert in begining
2.Insert at last
3.Show
4.Exit

Enter your choice?
```

PROGRAM NO 9

AIM :Write a program of Link list implementation for deletion

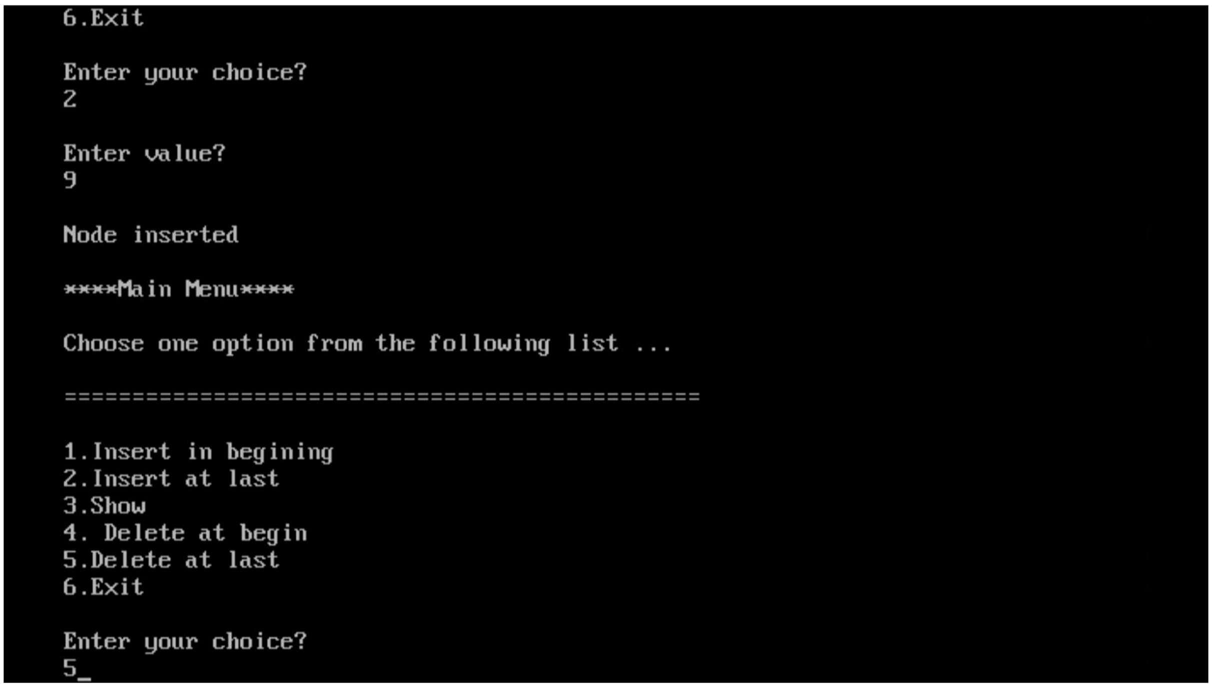
CODE :

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
};
struct node *head;

void beginsert ();
void lastinsert ();
void display();
void begin_delete();
void last_delete();

void main ()
{
    int choice =0;
    while(choice != 6)
    {
        printf("\n\n****Main Menu****\n");
        printf("\nChoose one option from the following list ...\n");

printf("\n===============================================\n");
        printf("\n1.Insert in begining\n2.Insert at last\n3.Show\n4.
Delete at begin\n5.Delete at last\n6.Exit\n");
        printf("\nEnter your choice?\n");
        scanf("\n%d",&choice);
        switch(choice)
        {
            case 1:
            beginsert();
            break;
            case 2:
            lastinsert();
            break;
            case 3:
            display();
            break;
  case 4:
  begin_delete();
  break;
  case 5:
  last_delete();
  break;
            case 6:
            exit(0);
            break;
            default:
```

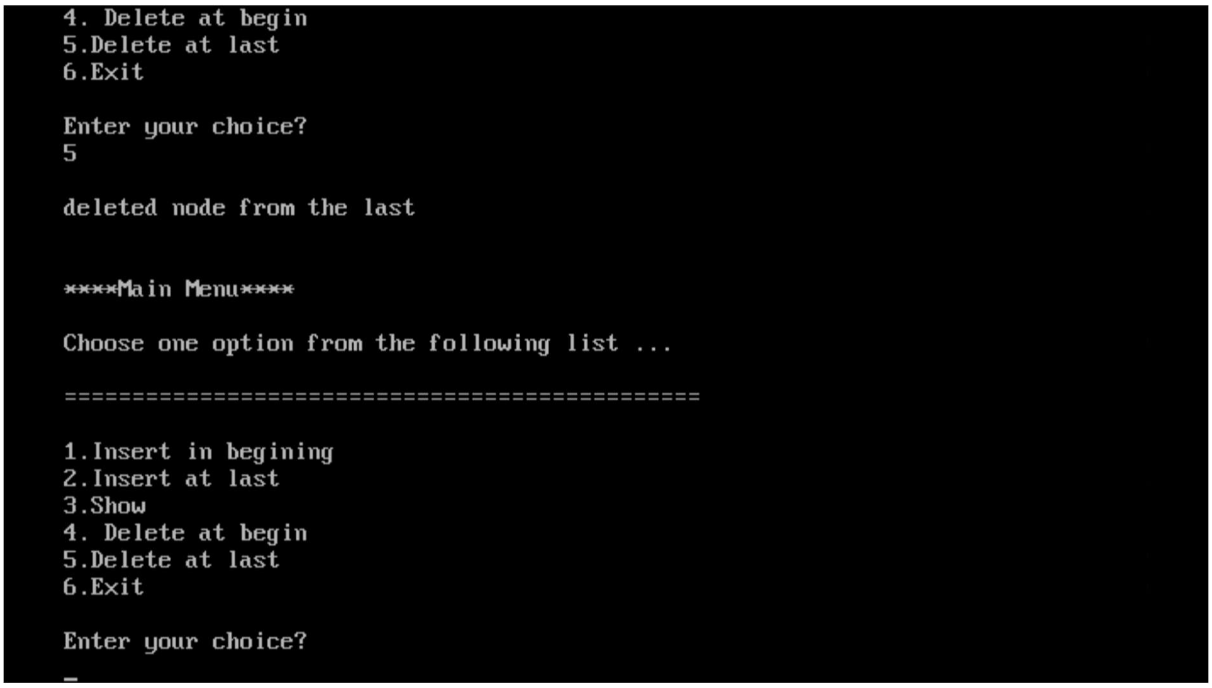
```
            printf("Please enter valid choice..");
        }
    }
}
void beginsert()
{
    struct node *ptr;
    int item;
    ptr = (struct node *) malloc(sizeof(struct node *));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter value\n");
        scanf("%d",&item);
        ptr->data = item;
        ptr->next = head;
        head = ptr;
        printf("\nNode inserted");
    }

}
void lastinsert()
{
    struct node *ptr,*temp;
    int item;
    ptr = (struct node*)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter value?\n");
        scanf("%d",&item);
        ptr->data = item;
        if(head == NULL)
        {
            ptr -> next = NULL;
            head = ptr;
            printf("\nNode inserted");
        }
        else
        {
            temp = head;
            while (temp -> next != NULL)
            {
                temp = temp -> next;
            }
            temp->next = ptr;
            ptr->next = NULL;
            printf("\nNode inserted");

        }
    }
}

void display()
{
    struct node *ptr;
    ptr = head;
    if(ptr == NULL)
    {
        printf("Nothing to print");
    }
    else
    {
        printf("\nprinting values . . . . .\n");
        while (ptr!=NULL)
        {
            printf("\n%d",ptr->data);
            ptr = ptr -> next;
        }
    }
}

void begin_delete()
{
     struct node *ptr;
     if(head==NULL)
{
     printf("\n list is empty\n");
}
else
{
     ptr=head;
     head= ptr -> next ;
     free(ptr);
     printf("\n node deleted from beginning \n");
}
}
void last_delete()
{
     struct node *ptr,*ptr1;
     if(head==NULL)
     {
          printf("\n list is empty\n");
     }
     else if(head -> next == NULL)
     {
          head=NULL;
          free(head);
          printf("\n only node of the list deleted \n");
     }
     else
     {
          ptr=head;
          while(ptr-> next!=NULL)
     {
          ptr1=ptr;
```

```
                ptr=ptr->next;
            }
                ptr1->next=NULL;
                free(ptr);
                printf("\ndeleted node from the last \n");
            }
        }
```

OUTPUT :



```
6.Exit

Enter your choice?
2

Enter value?
9

Node inserted

****Main Menu****

Choose one option from the following list ...

===============================================

1.Insert in begining
2.Insert at last
3.Show
4. Delete at begin
5.Delete at last
6.Exit

Enter your choice?
5_
```



```
4. Delete at begin
5.Delete at last
6.Exit

Enter your choice?
5

deleted node from the last


****Main Menu****

Choose one option from the following list ...

===============================================

1.Insert in begining
2.Insert at last
3.Show
4. Delete at begin
5.Delete at last
6.Exit

Enter your choice?
_
```

PROGRAM NO 10

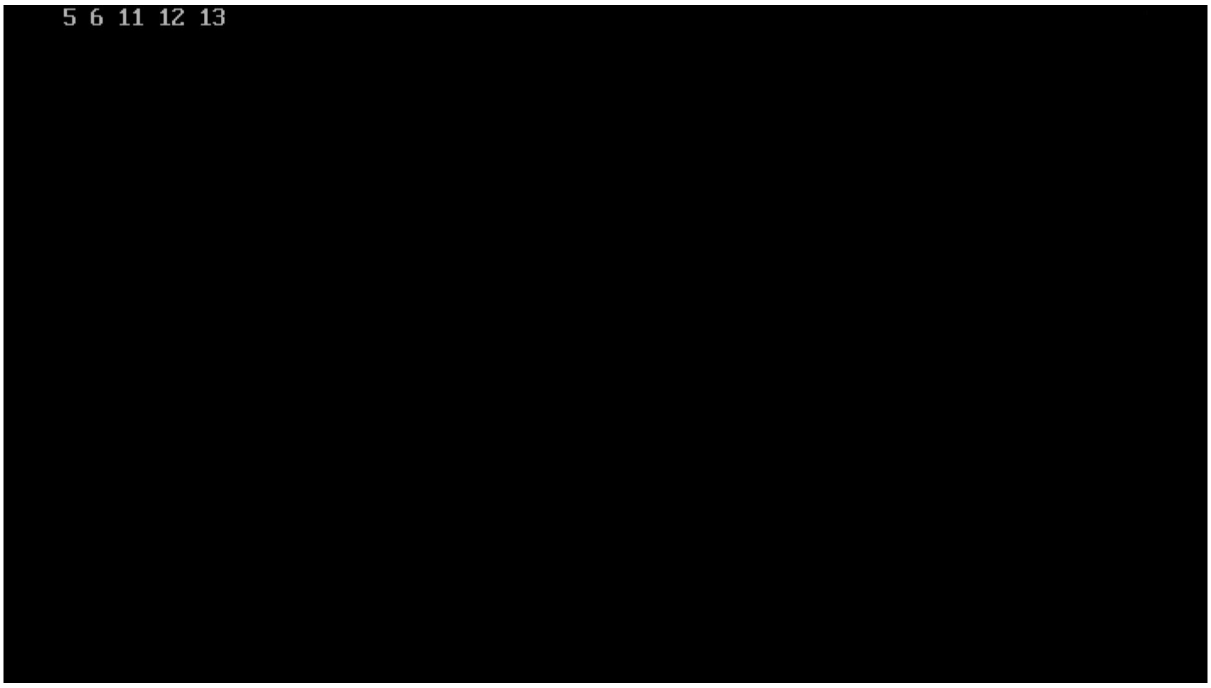
AIM :Write a program of Insertion Sort

CODE :

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

/* Function to sort an array
   using insertion sort*/
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i - 1;

        /* Move elements of arr[0..i-1],
           that are greater than key,
           to one position ahead of
           their current position */
        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

// A utility function to print
// an array of size n
void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

// Driver code
```

```
void main()
{
   int arr[] = {12, 11, 13, 5, 6};
   int n = sizeof(arr) / sizeof(arr[0]);

   insertionSort(arr, n);
   printArray(arr, n);

   getch();
}
```

OUTPUT :



```
5 6 11 12 13
```