Ex.No:1a
Date:

Insert a new element in array

Aim:

Program to insert a new element and print the position of a particular element

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

- Start
- Declare the variables
- Read N, a[i],item,k
- Shift the elements until the desired position is found
- Place the item
- Print the result
- Stop

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[20],i,n,k,item;
clrscr();
printf("enter the size n of the array\n");
scanf("%d",&n);
printf("enter the elements one by one\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
printf("the array elements before inserting a new element is\n");
for(i=0;i<n;i++)
printf("\n%d",a[i]);
printf("\n enter the new item to insert\n");
scanf("%d",&item);
printf("enter the position to insert between 0 and n\n");
scanf("%d",&k);
for(i=k;i>k;i--)
a[i+1]=a[i];
a[k]=item;
n=n+1;
printf("the array elements after inserting a new element is\n");
for(i=0;i<n-1;i++)
printf("\n%d",a[i]);
getch();
}
```

Output

```
enter the size n of the array
3
enter the elements one by one
54
67
34
the array elements before inserting a new element is
54
67
34
enter the new item to insert
87
```

```
enter the position to insert between 0 and n
1
the array elements after inserting a new element is
54
87
34
```

Result

Thus the program insertion of new element at the position of a particular element is compiled, executed and the required output is obtained.

Ex.No:1b

Date:

Delete an element from array

Aim:

Program to delete an element from an array and print the position of a particular element.

Hardware Requirements: Computer with Pentium IV Processor

Software Requirements: Turbo C Compiler With Editor

Algorithm:

- Start
- Declare the variables
- Read N, a[i],item,k
- Copy the elements to be deleted from array to a variable, place the item
- Shift the elements
- print the result
- Stop

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[20],i,n,k,item;
clrscr();
printf("enter the size n of the array\n");
scanf("%d",&n);
printf("enter the elements one by one\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
printf("the array elements before deleting and element is\n");
for(i=0;i<n;i++)
printf("\n%d",a[i]);
printf("\nenter the position to delete between 0 and n-1\n");
scanf("%d",&k);
for(i=k;i<n;i++)
a[i]=a[i+1];
n=n-1;
printf("the array elements after deleting an element is \n");
for(i=0;i<n;i++)
printf("\n%d",a[i]);
getch();
}
```

Output

```
enter the size n of the array
```

3
enter the elements one by one
56
99
12
the array elements before deleting and element is
56
99
12
enter the position to delete between 0 and n-1
1
the array elements after deleting an element is
56
12

Result Thus the program delete an element from array is compiled, executed and the required output is obtained

Ex.No:2
Date:

ROW MAJOR AND COLUMN MAJOR

Aim:

Program to implement array using row major order and column major order.

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.Start

2.declare the variables

3.Get the total no of columns and rows.

4.Arrange the values in rows and columns format.

5.Print the result.

6.stop

Program

```
#include<stdio.h>
Define MAX 10
Void main()
{
Int arr[max][max],m,n,i,j;
Printf("enter total number of rows");
Scanf("%d",&m);
Printf("enter total number of columns");
Scanf("%d",&n);
For(i=0;i<m;i++)
{
For(j=0;j<n;j++)
```

```
{
Printf(“enter numbers”);
Scanf(“%d”,&arr[i][j]);
}
}
Printf(“row major order\n”);
For(i=0;i<m;i++){
For(j=0;j<n;j++){
Printf(%d”,arr[i][j]);
}
}
Printf(“column  major order\n”);
For(i=0;i<m;i++){
For(j=0;j<n;j++){
Printf(%d”,arr[j][i]);
}
}
}
```

```
enter total number of rows
2
enter total number of columns
2

enter numbers
1
enter numbers
2
enter numbers
3
enter numbers
4

row major order1234

column  major order1324
```

Result

Thus the program implement array row major and column major is compiled, executed and the required output is obtained

Ex.No:3
Date:

Two Dimensional Array

Aim:

To create a two dimensional array with atleast ten elements. To search for a particular element and print its position and address of the element.

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.Start

2.declare the variables

3.read number of rows and columns

4.Read a[i][j],item

5.check if a[i][j]=item, if it is true find the position of the search item else search item is not present

6.print the result

7.stop

Program

```
#include<stdio.h>
#include<conio.h>
void main()
```

```
{
int a[20][20],i,j,m,n,pos,item;
char f;
int*p=&a[0][0];
clrscr();
printf("enter the number of row\n");
scanf("%d",&m);
printf("enter the number of column\n");
scanf("%d",&n);
printf("enter the data one by one in row wise\n");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
scanf("%d",&a[i][j]);
}
printf("enter the search item\n");
scanf("%d",&item);
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
if(a[i][j]==item)
{
p=p+(i*n+j)*16;
f='y';
printf("the search item %d iS in the position %d,%d",item,i,j);
printf("\n the address of the search item %dis%d",item,p);
break;
}
}
}
if(f!='y')
printf("\n The search item %d is not present in the array",item);
getch();
}
```

OUTPUT

```
enter the number of row
2
enter the number of column
```

2

enter the data one by one in row wise

3

5

1

6

enter the search item

5

the search item 5 iS in the position 0,1

Result Thus the program creation of two dimensional array is compiled, executed and the required output is obtained

Ex.No:4
Date:

Stack operations using Array

Aim:

To perform operations in stack by using array

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.Start
2.declare a structure stack,members and structure variable.
3.initialize stack size n
4.initialize stack member s.top=-1
5.read opt

6.if opt=1 then top is incremented to push element
7.if opt=2 then top is decremented to pop element
8.if opt=3 then display stack elements
9.if opt=4 then exit from the menu
10.stop

Program

```
#include<stdio.h>
#include<conio.h>
#define n 10
struct stack
{
int top;
int data[n];
};
void main()
{
struct stack s;
int opt,item,i;
s.top=-1;
clrscr();
do
{
printf("\n1.Push\n");
printf("2.Pop\n");
printf("3.List\n");
printf("4.Exit\n");
printf("Enter any one option\n");
scanf("%d",&opt);
switch(opt)
{
case 1:if(s.top==n-1)
printf("Stack Full\n");
else
{
printf("Enter the data to push in to the stack\n");
scanf("%d",&item);
s.top=s.top+1;
s.data[s.top]=item;
}
break;
case 2:if(s.top==-1)
printf("Stack empty\n");
else
{
printf("the deleted element from stack is %d",s.data[s.top]);
s.top--;
}
break;
case 3:if(s.top==-1)
printf("Stack empty\n");
else
{
printf("The elements in the stack are\n");
for(i=s.top;i>=0;i--)
printf("\n%d\n",s.data[i]);
}
break;
```

```
}
}while(opt!=4);
getch();
}
```

OUTPUT

```
1.Push
2.Pop
3.List
4.Exit;
Enter any one option
1

Enter the data to push in to the stack
33
1.Push
2.Pop
3.List
4.Exit;
Enter any one option
1

Enter the data to push in to the stack
22
1.Push
2.Pop
3.List
4.Exit;
Enter any one option
3

The elements in the stack are
22
33
```

Result

Thus the program creation of two dimensional array is compiled, executed and the required output is obtained

Ex.No:5
Date:

Reverse the string using stack

Aim:

Program to display the reverse of string using a stack.

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare a variables
3.check whether stack is full or not.
4.Push the string into stack.
5.Pop the string from the stack .
6.print the result.
7.Stop

Program

```
#include<stdio.h>
#include<string>
#define max 100
Int top,stack[max];
Void push(char  x);
{
If(top==max-1)
{
Printf("stack overflow");
}
Else
{
Stack(++top)=x;
}
}
Void pop();
Printf("%c",stack[top--]);
}
Main()
{
Char str()="program";
Int len=strlen(str).i;
For (i=0;i<len;i++)
Push(str[i]);
For(i=0;i<len;i++)
Pop();
}
```

OUTPUT
margorp

Result

Thus the program reverse the string using stack is compiled, executed and the required output is obtained

Ex.No:6
Date:

Evaluating Postfix Expression

Aim:

To evaluate a postfix expression using stack

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare a structure stack, members and variables
3.Declare character array exp1,opt,poniter variable *p
4.initialize s.top=-1
5.Get the postfix expression
6.scan the postfix expression characterwise from left to right till last character
7.if the character is operand push into the stack
8.if the character is an unary operator pop up one operand, if the operator is binary operator pop up two operands and perform the operation and push the result into the stack.
9.Repeat the steps 6,7,8 till end of the expression
10.Stop

Program

```
#include<stdio.h>
#include<math.h>
#include<ctype.h>
struct stack
{
int top;
double a[50];
};
void main()
{
char exp1[50],*p,opt;
struct stack s;
double d1,d2,d3;
clrscr();
s.top=-1;
printf("\n enter the postfix expression...");
gets(exp1);
p=exp1;
while(*p!='\0')
{ opt=*p;
switch(opt)
{
case '0':
case '1':
case '2':
case '3':
case '4':
case '5':
case '6':
case '7':
case '8':
case '9':
```

```
s.a[++s.top]=toascii(opt)-48;
break;
case '+':
d2=s.a[s.top--];
d1=s.a[s.top--];
s.a[++s.top]=d1+d2;
break;
case '-':
d2=s.a[s.top--];
d1=s.a[s.top--];
s.a[++s.top]=d1-d2;
break;
case '*':
d2=s.a[s.top--];
d1=s.a[s.top--];
s.a[++s.top]=d1*d2;
break;
case '/':
d2=s.a[s.top--];
d1=s.a[s.top--];
s.a[++s.top]=d1/d2;
break;
case '^':
d2=s.a[s.top--];
d1=s.a[s.top--];
d3=exp1[d2*log(d1)];
s.a[++s.top]=d3;
break;
}
p++;
}
printf("\n the value of the expressions...%7f",s.a[s.top]);
getch();
}
```

OUTPUT

```
enter the postfix expression...65+
the value of the expressions...11
```


Result

Thus the program to evaluate a postfix expression is compiled, executed and the required output is obtained.

Ex.No:7
Date:

QUEUE

Aim:

To create a queue containing ten elements

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare a structure queue, members and variables
3.initialize q.front=-1, q.rear=-1
4.if opt=1 then rear pointer is incremented to add element
5.if opt=2 then front pointer is incremented to delete element
6.scan the postfix expression characterwise from left to right till last character
7.if the character is operand push into the stack
8.if the character is an unary operator pop up one operand, if the operator is binary operator pop up two operands and perform the operation and push the result into the stack.
9.Repeat the steps 6,7,8 till end of the expression
10.Stop

Program

```
#include<stdio.h>
#include<conio.h>
struct queue
{
int data[50];
int front;
int rear;
};
void main()
{
struct queue q;
int i,x,opt,item;
q.front=-1;
q.rear=-1;
clrscr();
do
{
printf("selected any one option\n");
printf("1.create\n");
printf("2.list\n");
printf("3.insert\n");
printf("4.delete\n");
printf("5.exit\n");
scanf("%d",&opt);
switch(opt)
{
case 1:
printf("enter 5 data one by one\n");
for(i=0;i<10;i++)
scanf("%d",&q.data[++q.rear]);
break;
case 2:
```

```
printf("the data in the queue are\n");
for(i=q.front+1;i<=q.rear;i++)
printf("%d\n",q.data[i]);
break;
case 3:
printf("enter the data to insert\n");
scanf("%d",&item);
q.data[++q.rear]=item;
break;
case 4:
x=q.data[q.front];
printf("\n the deleted item is \t%d\n",x);
break;
}
}while(opt!=5);
getch();
}
```

Output

```
selected any one option
1.create
2.list
3.insert
4.delete
5.exit

1
enter 5 data one by one
7
5
12
56
78
selected any one option
1.create
2.list
3.insert
4.delete
5.exit
2
the data in the queue are
7
5
12
56
78
```

Result

Thus the program to create a queue containing ten elements is compiled,executed and required output is obtained.

Ex.No:8
Date:

Recursive function

Aim:

Program to implement recursive function.

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.Start.
2.Declare the variable.
3.Using recursion find the factorial value.
4.Use (n*fact(n-1)) to calculate factorial.
5.Print the result.
6.Stop.

Program

```
  #include<stdio.h>
Int fact(int);
Int main()
{
Int num,f;
Printf(“enter a number”);
Scanf(“%d”,&num);
F=fact(num);
Printf(“factorial of %d is:%d”,num,f);
Return 0;
}
Int fact(int n)
{
If(n==1)
Return 1;
Else
```

```
Return(n*fact(n-1));
}
```

OUTPUT
enter a number
4
factorial of 4 is:24

Result

Thus the program recursive function is compiled,executed and the required output is obtained.

Ex.No:9
Date:

Singly Linked List

Aim:

To create a singly linked list containing at least five elements

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.Start
2.Declare a structure to define a node
3.Get new node using getnode() function
4.A function name create() to store the value and the address of the next data in the empty node.
5.function name list() to print the data stored in the linked list
6.End

Program

```
#include<stdio.h>
#include<malloc.h>
struct node
{
int p,data;
struct node * link;
};
struct node * getnode()
{
struct node * p;
p=(struct node *)malloc(sizeof(struct node));
return(p);
}
struct node *create()
{
struct node * ptr,* newl,*prev;
```

```
int i,r,n;
clrscr();
ptr=getnode();
newl=ptr;
printf("\n enter the number of data to store \n");
scanf("%d",&n);
printf("\n enter the data one by one \n");
for(i=0;i<n;i++)
{
scanf("%d",&r);
newl->data=r;
prev=newl;
newl=getnode();
prev->link=newl;
}
prev->link=0;
return(ptr);
}
void list(p)
struct node * p;
{
struct node * q;
q=p;
printf("\n the list contains \n");
while(q!=0)
{
printf("\n %d",q->data);
q=q->link;
}
}
void main()
{
struct node * p;
p=create();
list(p);
getch();
}
```

OUTPUT

```
enter the number of data to store

5

enter the data one by one
40
20
10
30
50

the list contains

40
20
10
30
```

Result

Thus the program creation of a singly linked list is compiled,executed and the required output is obtained.

Ex.No:10
Date:

Delete first node in Singly Linked List

Aim:

To delete the first node that contains an integer data item of a singly linked list

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare a structure to define a node
3.Get new node using getnode() function
4.a function name create() to store the value and the address of the next data in the empty node
5.replace the pointer of the linked list p with the second node address p->plink
6.use free() function to free the deleted node
7.a function name list() to print the data stored in the linked list
8.end

Program

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
struct node
{
int data;
```

```
struct node*link;
};
void main()
{
struct node*getnode();
struct node*create();
struct node*del(struct node*);
void list(struct node*);
struct node*p;
clrscr();
p=create();
printf("the data in the list are\n");
list(p);
p=del(p);
printf("\n the data in the list after deleting first data is\n");
list(p);
getch();
}
struct node *getnode()
{
struct node *p;
p=(struct node*)malloc(sizeof(struct node));
return(p);
}
struct node*create()
{
struct node*ptr,*newl,*prev;
int i,r,n;
ptr=getnode();
newl=ptr;
printf("\n enter the maximum number of data to add\n");
scanf("%d",&n);
printf("enter the data one by one\n");
for(i=0;i<n;i++)
{
scanf("%d",&r);
newl->data=r;
prev=newl;
newl=getnode();
prev->link=newl;
}
prev->link=0;
return(ptr);
}
struct node *del(struct node *p)
{
p=p->link;
return(p);
}
void list(struct node *p)
{
struct node*q;
q=p;
while(q!=0)
{
printf("\n%d",q->data);
q=q->link;
}
}
```

OUTPUT
enter the maximum number of data to add
4

enter the data one by one
3
4
6
8

the data in the list are

3
4
6
8

the data in the list after deleting first data is

4
6
8

Result

Thus the program delete the first node in the singly linked list is created,executed and the required output is obtained.

Ex.No:11
Date:

Binary Tree

Aim:

To create a binary tree using c

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare a structure node, define the structure members
3.get a new node using getnode( ) function
4.a function name create ( ) to store the data value in the data field, address of the left child node in lchild field and the address of the right child node in rchild field. Print the value of left and right child in a binary tree

5.End

Program

```
#include<stdio.h>
#include<conio.h>
struct node
{
int data;
struct node * lchild;
struct node * rchild;
};
struct node * getnode()
{
struct node * p;
p=(struct node *)malloc (sizeof(struct node));
return p;
}
struct node * create()
{
struct node *ptr,*t,*t1[100];
int n,i,value,k=1;
clrscr();
printf("\n enter the number of data to insert ..");
scanf("%d",&n);
printf(" enter the data one by one\n");
scanf("%d",&value);
ptr=getnode();
ptr->data=value;
ptr->lchild=NULL;
ptr->rchild=NULL;
t1[k++]=ptr;
for(i=2;i<=n;i++)
{
scanf("%d",& value);
t=getnode();
t->data=value;
t1[k++]=t;
if(i%2==0)
{
t1[i/2]->lchild=t;
printf("\n left child %dis created",i);
t->lchild=NULL;
t->rchild=NULL;
}
else
{
t1[i/2]->rchild=t;
printf("\n right child %d is created",i);
t->lchild=NULL;
t->rchild=NULL;
}
}
return(ptr);
}
void main()
{
clrscr();
create();
getch();
```

}
OUTPUT
enter the number of data to insert ..
5
enter the data one by one
23
11
left child 2 is created
45
right child 3 is created
60
left child 4 is created
38
right child 5 is created

Result

Thus the program binary tree is created, compiled executed and the required output is obtained.

Ex.No:12
Date:

Preorder Traversal of a Binary Tree

Aim:

To create a Preorder traversal binary tree using c

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare a structure node, define the structure members
3.get a new node using getnode( ) function
4.a function name create ( ) to store the data value in the data field, address of the left child node in lchild field and the address of the right child node in rchild field.
5.start from the root node and visit this node print the data,
6.Move left, print the data and this process is continued till there is no path to move.
7. Move right, print the data and move left print the data and this process is continued till there is no path to move.
8. move one node back and do step 7
9.End

Program

```
#include<stdio.h>
#include<conio.h>
struct node
{
int data;
struct node * lchild;
struct node * rchild;
};
struct node * getnode()
{
struct node * p;
p=(struct node *)malloc (sizeof(struct node));
return p;
}
struct node *create()
{
struct node *ptr,*t,*t1[100];
int n,i,value,k=1;
clrscr();
printf("\n enter the number of data to insert...");
scanf("%d",&n);
printf("\n enter the data one by one \n");
scanf("%d",&value);
printf("Root node is create\n");
ptr=getnode();
ptr->data=value;
ptr->lchild=NULL;
ptr->rchild=NULL;
t1[k++]=ptr;
for(i=2;i<=n;i++)
{
scanf("%d",&value);
t=getnode();
t->data=value;
t1[k++]=t;
if(i%2==0)
{
t1[i/2]->lchild=t;
printf("\n left child of %d is created",i);
t->lchild=NULL;
t->rchild=NULL;
}
else
```

```
{
t1[i/2]->rchild=t;
printf("\n right child of %d is created",i);
t->lchild=NULL;
t->rchild=NULL;
}
}
return(ptr);
}
void preorder (struct node*p)
{
if(p!=NULL)
{
printf("%d\t",p->data);
preorder(p->lchild);
preorder(p->rchild);
}
}
void main()
{
struct node *p;
clrscr();
p=create();
preorder(p);
getch();
}
```

OUTPUT

```
enter the number of data to insert...
4
enter the data one by one
55
Root node is create
44
left child of 2 is created
62
right child of 3 is created
5
left child of 4 is created
```

Result

Thus the program preorder traversal of a binary tree is created, compiled,executed and the required output is obtained.

Ex.No:13

Date:

Binary Search

Aim:

To create a program in 'C' for binary search

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare the variables
3.let a be an array of sorted data
4.let x be the given data for searching
5.find the mid position of the given array as mid=lb+ub/2
6.compare x with mid position value. If it equals searching over print search success.
7.if x is less than mid value, repeat steps 4,5 for the sub array 0 to mid-1
8.if x is greater than mid value, repeat steps 4,5 for the sub array mid+1 to lb
9.if no value matches print search failed.
10.end

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
int lb,ub,a[10],mid,x,n,i;
clrscr();
printf("\n Enter the maximum number of elements \n");
scanf("%d",&n);
printf("Enter the elements one by one \n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
printf("Enter the element to search \n");
scanf("%d",&x);
lb=0;
ub=n-1;
while(lb<=ub)
{
mid=(lb+ub)/2;
if(a[mid]==x)
{
printf("\n The element is present at %d positon",mid);
break;
}
else
if(a[mid]>x)
ub=mid-1;
else
lb=mid+1;
}
if(lb>ub)
printf("\n search failed");
getch();
```

}

Output
Enter the maximum number of elements
5
Enter the elements one by one
22
44
32
89
11

Enter the element to search
11

The element is present at 4 positon

Result

Thus the program binary search is compiled, executed and the required output is obtained.

Ex.No:14
Date:

Insertion Sort

Aim:

To write a program in c to sort n numbers using insertion sort

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare the variables and array
3.read the variable and array
4.to use nested for loop count and compare the array data
5.Scan the array a from starting to n and find the smallest elements and insert into its proper position in the previously sorted sub array.
5.print the sorted data
6.stop

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,x,k,n,a[20];
clrscr();
printf("give the number of elements \n");
scanf("%d",&n);
printf("give the elements one by one \n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=0;i<n;i++)
{
x=a[i];
j=i;
while(x<a[j-1]&&j-1>=0)
j--;
for(k=i;j<k;k--)
a[k]=a[k-1];
a[j]=x;
}
printf("\nthe sorted list is \n");
for(i=0;i<n;i++)
printf("%d\n",a[i]);
getch();
}
```

Output
give the number of elements
4
give the elements one by one
87
34
76
12
the sorted list is
12
34
76
87

Result: **Thus the program insertion sort is compiled executed and the required output is obtained.**

Ex.No:15
Date:

Bubble Sort

Aim:

To write a program in 'C' to sort N numbers using bubble sort

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare the array and variables
3.take the first element and compare with the second element. If it is small don't interchange or else interchange the two elements
4.then compare the second element with the third element. if it is small don't interchange or else interchange the two elements.
5.this process is continued till (n-1)th element is compared with the nth element.
6.at the end of the first scan, the highest element comes to the nth position.
7.repeat steps 3,4,5 till the element is come to the sorted order.
8.end

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,n,a[20],temp;
clrscr();
printf("\n Give the number of data");
scanf("%d",&n);
printf("Give the data one by one \n");
```

```
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=1;i<n;i++)
for(j=0;j<n-1;j++)
{
if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
printf("\n The sorted list \n");
for(i=0;i<n;i++)
printf("%d\n",a[i]);
getch();
}
```

OUTPUT

Give the number of data
5
Give the data one by one
9
1
4
7
2
The sorted list
1
2
4
7
9

Result Thus the program is compiled, executed and the required output is obtained.

Ex.No:16
Date:

Selection Sort

Aim:

To write a program in 'C' to sort N numbers using selection sort

Hardware Requirements

Computer with Pentium IV Processor

Software Requirements

Turbo C Compiler With Editor

Algorithm:

1.start
2.Declare the array and variables
3.take the first element and compare with the second element. If it is small don't interchange or else interchange the two elements
4.then compare the second element with the third element. if it is small don't interchange or else interchange the two elements.
5.this process is continued till (n-1)th element is compared with the nth element.

6.at the end of the first scan, the highest element comes to the nth position.
7.repeat steps 3,4,5 till the element is come to the sorted order.
8.end

Program

```
#include<stdio.h>
Int main()
{
Int s,i,j,temp,a[20];
Printf("enter total elements");
Scanf("%d",&s);
Printf("enter %d elements;");
For(i=0;i<s;i++)
Scanf("%d",&a[i]);
For(i=0;i<s;i++)
{
For(j=i+1;j<s;j++)
{
Ifa[i]>a[j])
{
Temp=a[i];
A[i]=a[j];
A[j]=temp;
}
}
}printf("after sorting ");
For(i=0;i<s;i++)
Printf("&d",a[i]);
}
```

OUTPUT

```
 enter total elements
 5
Enter 5 elements
2
4
9
1
6
After sorting12469
```

Result: **Thus the program selection sort is compiled, executed and the required output is obtained.**