LAB FILE

Data Structures using 'C' Lab

(BSIT-C251)

Bachelor of Science (Information Technology)

Department of Computer Application

Uttaranchal Institute of Management

Under the Supervision of:

Ishteyaaq Ahmad

(Assistant Professor)

Department of Computer Application

Submitted by:

Vansh Kumar

2010000061

BSc. IT II Sem

Session 2020-21

INDEX

PRACTICAL 1

Design, Develop and Implement a menu driven Program in C for the following Array operations

a. Creating an Array of N Integer Elements
b. Display of Array Elements with Suitable Headings
c. Inserting an Element (ELEM) at a given valid Position (POS)
d. Deleting an Element at a given valid Position (POS)
e. Exit.
Support the program with functions for each of the above operations.

PROGRAM: -

```
#include<stdio.h>
#include<process.h>
void create();void insert();void del();void head();
int n,i,pos,val;
int arr[100];
int main()
{
int ch;
while(1)
{
printf("\n\nThis is a menu driven program for array.\n");
printf("***************************************************************\n");
printf("Enter 1 to create an array.\n");
printf("Enter 2 to create array elements with suitable heading.\n");
printf("Enter 3 to insert element at given valid position.\n");
printf("Enter 4 to delete element from given valid position.\n");
printf("Enter 5 to exit program.\n");
printf("***************************************************************\n\n");
printf("Enter any of above options:- ");
scanf("%d",&ch);
switch(ch)
{
case 1:
create();
break;
case 2:
create();
head();
break;
case 3:
create();
```

```
insert();
break;
case 4:
create();

del();
break;
case 5:
exit (0);
default :
printf("\n OOPS, you have entered an invalid choice.\n\n");
}
}
return 0;
}
void create()
{
printf("\n Enter the size of array:- ");
scanf("%d",&n);
printf("\n Enter the array elements:- ");
for(i=0;i<n;i++)
{
scanf("\n %d",&arr[i]);
}
printf("\n The array elements are:- ");
for(i=0;i<n;i++)
printf(" %d",arr[i]);
}
void head()
{
printf("\n Array with suitable headings:- \n");
for(i=0;i<n;i++)
{
printf("\n arr[%d]= %d",i+1,arr[i]);
}
}
void insert()
{
printf("\n\n Enter the position at which you want to insert an element:- ");
scanf("%d",&pos);
if(pos>=n+1)
printf("\n Insertion isn't possible here.");
else
{
printf("\n Enter the element you want to insert:- ");
scanf("%d",&val);
```

```
for(i=n-1;i>=pos-1;i--)
arr[i+1]=arr[i];
arr[pos-1]=val;
printf("\n The new array is:- \n");
for(i=0;i<=n;i++)
printf("%d ",arr[i]);
}
}
void del()
{
printf("\n\n Enter the postion of element you want to delete:- ");
scanf("%d",&pos);
if(pos>=n+1)
printf("ERROR!!!!\n Deletion isn't possible.");
else
{
for(i=pos-1;i<n-1;i++)
arr[i]=arr[i+1];
printf("\n New array is:- \n");
for(i=0;i<n-1;i++)
printf("%d ",arr[i]);
}
}
```

OUTPUT: -

```
Enter any of above options:- 4

 Enter the size of array:- 6

 Enter the array elements:- 1
2
8
3
4
5

 The array elements are:-    1   2   8   3   4   5

 Enter the postion of element you want to delete:- 3

 New array is:-
1    2    3    4    5

This is a menu driven program for array.
************************************************************
Enter 1 to create an array.
Enter 2 to create array elements with suitable heading.
Enter 3 to insert element at given valid position.
Enter 4 to delete element from given valid position.
Enter 5 to exit program.
************************************************************

Enter any of above options:-
```

```
Enter any of above options:- 2

 Enter the size of array:- 5

 Enter the array elements:- 1
2
3
4
5

 The array elements are:-    1   2   3   4   5
 Array with suitable headings:-

 arr[1]= 1
 arr[2]= 2
 arr[3]= 3
 arr[4]= 4
 arr[5]= 5

This is a menu driven program for array.
************************************************************
Enter 1 to create an array.
Enter 2 to create array elements with suitable heading.
Enter 3 to insert element at given valid position.
Enter 4 to delete element from given valid position.
Enter 5 to exit program.
************************************************************

Enter any of above options:-
```

```
Enter any of above options:- 3

 Enter the size of array:- 4

 Enter the array elements:- 1
2
4
5

 The array elements are:-    1   2   4   5

 Enter the position at which you want to insert an element:- 3

 Enter the element you want to insert:- 3

 The new array is:-
1   2   3   4   5

This is a menu driven program for array.
************************************************************
Enter 1 to create an array.
Enter 2 to create array elements with suitable heading.
Enter 3 to insert element at given valid position.
Enter 4 to delete element from given valid position.
Enter 5 to exit program.
************************************************************

Enter any of above options:-
```

```
Enter 1 to create an array.
Enter 2 to create array elements with suitable heading.
Enter 3 to insert element at given valid position.
Enter 4 to delete element from given valid position.
Enter 5 to exit program.
*******************************************************

Enter any of above options:- 1

 Enter the size of array:- 5

 Enter the array elements:- 1
2
3
4
5

 The array elements are:-    1   2   3   4   5

This is a menu driven program for array.
*******************************************************
Enter 1 to create an array.
Enter 2 to create array elements with suitable heading.
Enter 3 to insert element at given valid position.
Enter 4 to delete element from given valid position.
Enter 5 to exit program.
*******************************************************

Enter any of above options:-
```

```
 Enter the size of array:- 6

 Enter the array elements:- 1
2
8
3
4
5

 The array elements are:-    1   2   8   3   4   5

 Enter the postion of element you want to delete:- 3

 New array is:-
1    2    3    4    5

This is a menu driven program for array.
*******************************************************
Enter 1 to create an array.
Enter 2 to create array elements with suitable heading.
Enter 3 to insert element at given valid position.
Enter 4 to delete element from given valid position.
Enter 5 to exit program.
*******************************************************

Enter any of above options:- 5

--------------------------------
Process exited after 175.6 seconds with return value 0
Press any key to continue . . .
```

PRACTICAL 2

Design, Develop and Implement a menu driven Program in C for the following Array operations

a. Creating an Array of N Integer Elements
b. Reverse the elements of array
c. Find maximum and minimum of array
d. Find even and odd elements of array
e. Find sum of elements of an array
f. Exit.
Support the program with functions for each of the above operations.

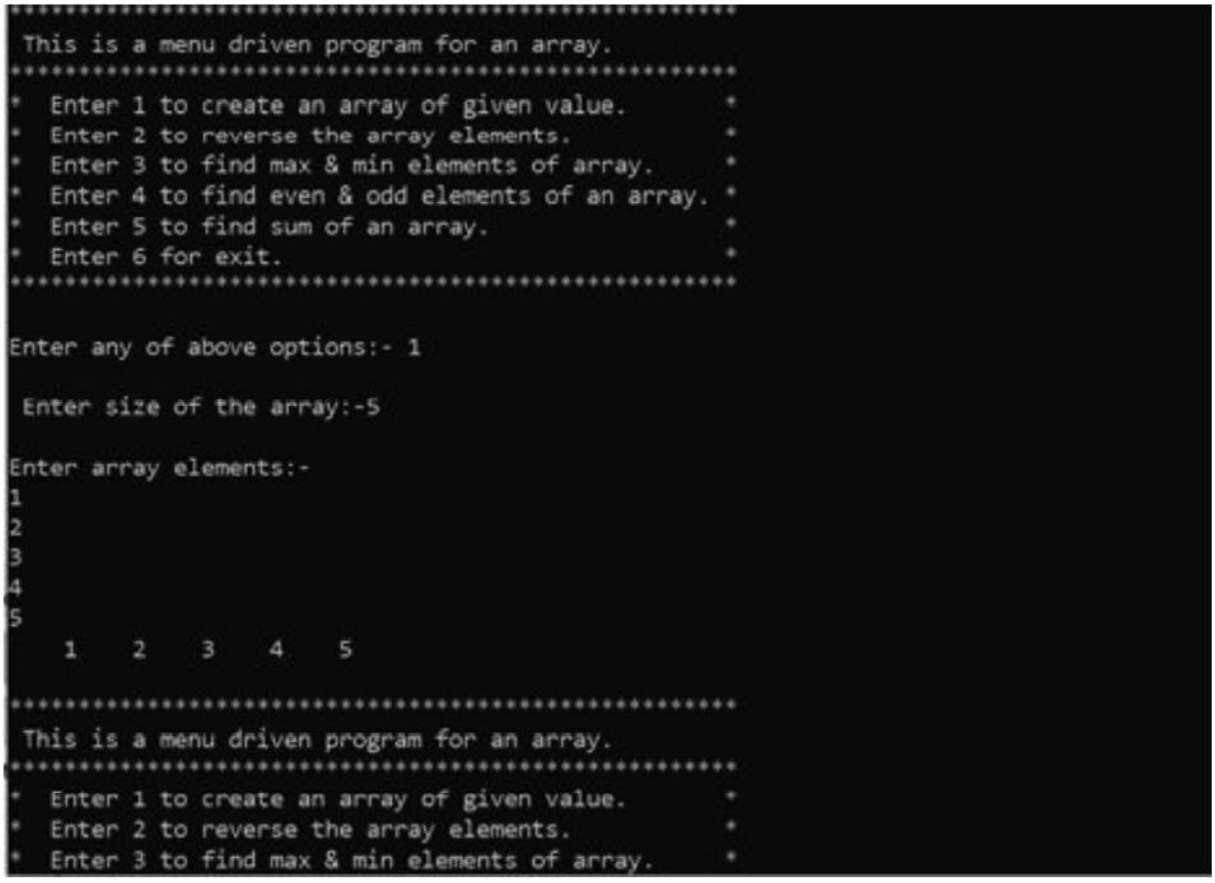
PROGRAM: -

```
#include<stdio.h>
#include<process.h>
void create( ); void reverse( ); void oddeven( ); void max( ); void sum( );
int i, n; int arr[50];
int main( )
{
int ch;
while(1)
{
printf("\n\n*********************************************************");
printf("\n This is a menu driven program for an array.\n");
printf("*********************************************************\n");
printf("* Enter 1 to create an array of given value. *\n");
printf("* Enter 2 to reverse the array elements. *\n");
printf("* Enter 3 to find max & min elements of array. *\n");
printf("* Enter 4 to find even & odd elements of an array. *\n");
printf("* Enter 5 to find sum of an array. *\n");
printf("* Enter 6 for exit. *\n");
printf("*********************************************************\n\n");
printf("Enter any of above options:- ");
scanf("%d",&ch);
switch(ch)
{
case 1:
create();
for (i=0;i<n;i++)
printf(" %d",arr[i]);
break;
```

```
case 2:
reverse();
break;
case 3:
max();
break;

case 4:
oddeven();
break;
case 5:
sum();
break;
case 6:
exit (0);
default :
printf("\n Oops, Invalid choice.");
}
}
return 0;
}
void create()
{printf("\n Enter size of the array:-");
scanf("\n %d",&n);
for(i=0;i<n;i++)
scanf("\n %d",&arr[i]);
}
void reverse()
{ create();
printf("\n Reverse of the array is:-");
for(i=n-1;i>=0;i--)
printf("\n %d",arr[i]);
}
void oddeven()
{ create();
printf("\n Odd and Even no are:-");
for(i=0;i<n;i++)
{
if(arr[i]%2!=0)
printf("\n Odd no:- %d",arr[i]);
else
printf("\n Even no:- %d",arr[i]);
}
}
void max()
{create();
```

```
int max,min;
max=arr[0];
for(i=0;i<=n;i++)
{
if(max<arr[i])
max=arr[i];
} printf("\n Maximum element is %d",max);
min=arr[0];
for(i=1;i<n;i++)
{
if(min>arr[i])
min=arr[i];
}
printf("\n Minimum element is %d",min);
}
void sum()
{int sum=0;
create();
for(i=0;i<=n;i++)
sum=sum+arr[i];
printf("\n Sum of the given array elements is:- %d",sum);
}
```

OUTPUT:



```
*  Enter 4 to find even & odd elements of an array. *
*  Enter 5 to find sum of an array.                  *
*  Enter 6 for exit.                                 *
*****************************************************

Enter any of above options:- 2

 Enter size of the array:-5

Enter array elements:-
1
2
3
4
5

 Reverse of the array is:-   5   4   3   2   1

*****************************************************
 This is a menu driven program for an array.
*****************************************************
*  Enter 1 to create an array of given value.       *
*  Enter 2 to reverse the array elements.           *
*  Enter 3 to find max & min elements of array.     *
*  Enter 4 to find even & odd elements of an array. *
*  Enter 5 to find sum of an array.                 *
*  Enter 6 for exit.                                *
*****************************************************

Enter any of above options:-
```

```
*  Enter 5 to find sum of an array.                 *
*  Enter 6 for exit.                                *
****************************************************

Enter any of above options:- 3

 Enter size of the array:-5

Enter array elements:-
19
68
10
33
98

 Maximum element is 98
 Minimum element is 10

****************************************************
 This is a menu driven program for an array.
****************************************************
*  Enter 1 to create an array of given value.      *
*  Enter 2 to reverse the array elements.          *
*  Enter 3 to find max & min elements of array.    *
*  Enter 4 to find even & odd elements of an array.*
*  Enter 5 to find sum of an array.                *
*  Enter 6 for exit.                               *
****************************************************

Enter any of above options:-
```

```
Enter any of above options:- 4

 Enter size of the array:-5

Enter array elements:-
1
5
9
66
35

 Odd and Even no are:-
 Odd no:- 1
 Odd no:- 5
 Odd no:- 9
 Even no:- 66
 Odd no:- 35

**************************************************
 This is a menu driven program for an array.
**************************************************
*  Enter 1 to create an array of given value.    *
*  Enter 2 to reverse the array elements.        *
*  Enter 3 to find max & min elements of array.  *
*  Enter 4 to find even & odd elements of an array. *
*  Enter 5 to find sum of an array.              *
*  Enter 6 for exit.                             *
**************************************************

Enter any of above options:-
```

```
*  Enter 4 to find even & odd elements of an array. *
*  Enter 5 to find sum of an array.              *
*  Enter 6 for exit.                             *
**************************************************

Enter any of above options:- 5

 Enter size of the array:-5

Enter array elements:-
14
8
32
4
18

 Sum of the given array elements is:- 76

**************************************************
 This is a menu driven program for an array.
**************************************************
*  Enter 1 to create an array of given value.    *
*  Enter 2 to reverse the array elements.        *
*  Enter 3 to find max & min elements of array.  *
*  Enter 4 to find even & odd elements of an array. *
*  Enter 5 to find sum of an array.              *
*  Enter 6 for exit.                             *
**************************************************

Enter any of above options:-
```

PRACTICAL 3

Design, Develop and Implement a menu driven Program in C for the following operations on two-dimensional array of Integers

a. Find addition of two matrix
b. Find transpose of a matrix
c. Find multiplication of two matrix
d. Find addition of two matrix
e. Determine given matrix is sparse or not.
f. Exit
Support the program with appropriate functions for each of the above operations

PROGRAM: -

```
#include<stdio.h>
#include<process.h>
#include<stdlib.h>
void add();void sub();void transpose();void sparse();
void data(int aa[ ][10],int bb[ ][10],int r1,int c1,int r2,int c2);
void multiply(int aa[ ][10],int bb[][10],int mul[ ][10],int r1,int c1,int r2,int c2);
void display(int mult[ ][10],int r1,int c2);
int aa[10][10],bb[10][10],mult[10][10],r1,c1,r2,c2;
int i,j,k,s=0,r,c,p,q,a[50][50],b[50][50],t[50][50],sum[50][50];
int main()
{
int ch;
while(1)
{
printf("\n\nThis is a menu driven program for matrix.\n");
printf("************************************************************\n");
printf("Enter 1 to add two matrices.\n");
printf("Enter 2 to transpose a matrix.\n");
printf("Enter 3 to multiply two matrices.\n");
printf("Enter 4 to subtract two matrices.\n");
printf("Enter 5 to determine wheather the given matrix is sparse or not.\n");
printf("Enter 6 to exit the program.\n");
printf("************************************************************\n\n");
printf("Enter any of above options:- ");
scanf("%d",&ch);
switch(ch)
{
case 1:
add();
break;
```

```
case 2:
transpose();
break;

case 3:
printf("\nEnter no. of rows and columns of first matrix:- \n");
scanf("%d %d",&r1,&c1);
printf("\nEnter no. of rows and columns of second matrix:- \n");
scanf("%d %d",&r2,&c2);
while(c1 != r2)
{
printf("\nERROR!!!\n\nEnter the no. of rows & columns of first matrix:- \n");
scanf("%d %d",&r1,&c1);
printf("\nEnter no. of rows & columns of second matrix:- \n");
scanf("%d %d",&r2,&c2);
}
data(aa,bb,r1,c1,r2,c2);
multiply(aa,bb,mult,r1,c1,r2,c2);
display(mult,r1,c2);
break;
case 4:
sub();
break;
case 5:
sparse();
break;
case 6:
exit (0);
default :
printf("\n OOPS YOU HAVE ENTERED AN INVALID OPTION.\n");
}
}
return 0;
}
void add()
{
printf("\nEnter the no. of rows of matrix:- ");
scanf("%d",&r);
printf("\nEnter the no. of columns:- ");
scanf("%d",&c);
printf("\nEnter elements of 1st matrix:-\n");

for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
```

```
printf("\nEnter element a[%d][%d]:- ",i+1,j+1);
scanf("%d",&a[i][j]);
}
}
printf("\nEnter elements of 2nd matrix:- \n");
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
printf("\nEnter element b[%d][%d]:- ",i+1,j+1);
scanf("%d",&b[i][j]);
}
}
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
sum[i][j]=a[i][j]+b[i][j];
}
}
printf("\nAddition of the matrices is:\n");
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
printf("%d ",sum[i][j]);
if(j== (c-1)){
printf("\n\n");
}
}
}
}

void sub()
{
printf("\nEnter the no. of rows of matrix:- ");
scanf("%d",&r);
printf("\nEnter the no. of columns:- ");
scanf("%d",&c);
printf("\nEnter elements of 1st matrix:-\n");
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
printf("\nEnter element a[%d][%d]:- ",i+1,j+1);
scanf("%d",&a[i][j]);
```

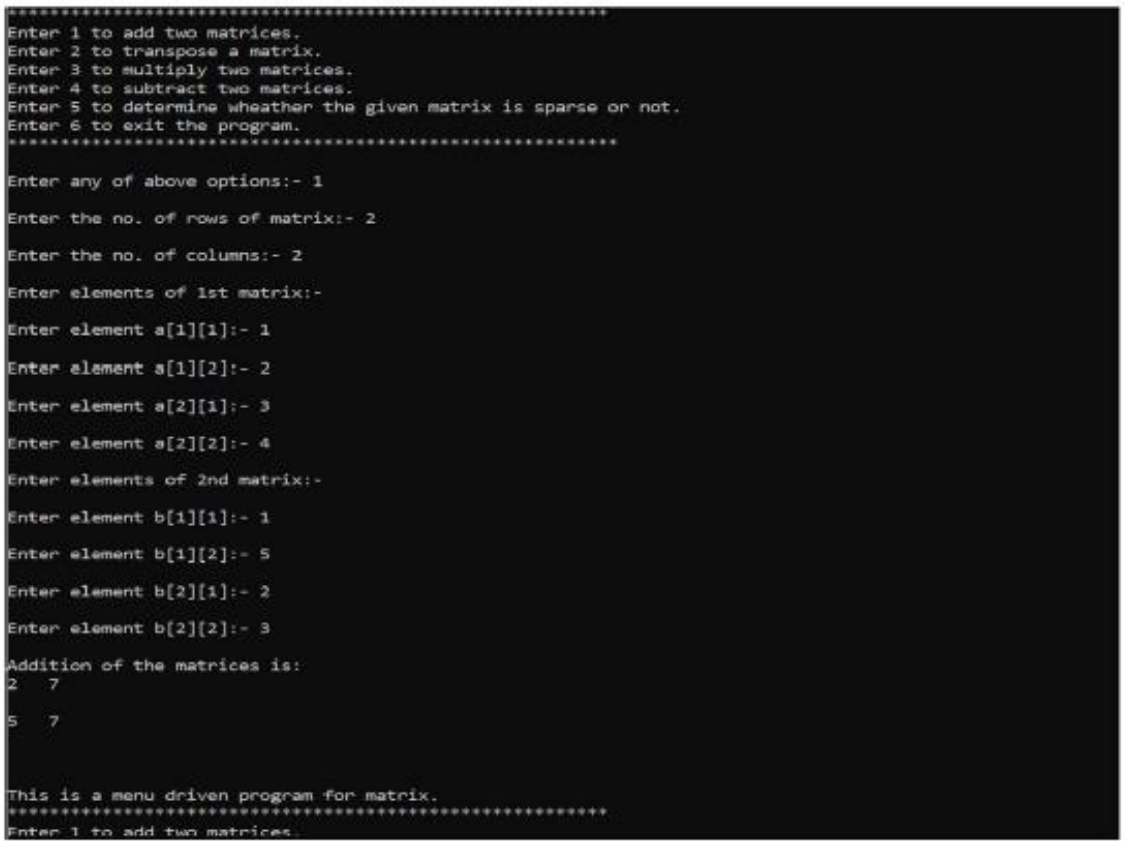
```
}
}
printf("\nEnter elements of 2nd matrix:- \n");
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
printf("\nEnter element b[%d][%d]:- ",i+1,j+1);
scanf("%d",&b[i][j]);
}
}
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
sum[i][j]=a[i][j]-b[i][j];
}
}
printf("\nSubtraction of the matrices is:\n");
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
printf("%d ",sum[i][j]);
if(j== (c-1)){
printf("\n\n");
}
}

} }
void transpose()
{
printf("\nEnter the number of rows:- ");
scanf("%d",&r);
printf("\nEnter the number of columns:- ");
scanf("%d",&c);
printf("\nEnter the elements of matrix:-\n");
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
printf("\nEnter element a[%d][%d]:-",i+1,j+1);
scanf("%d",&a[i][j]);
}
}
printf("\nThe given matrix is:-\n");
```

```
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
printf("%d ",a[i][j]);
if(j==(c-1))
printf("\n");
}
}
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
t[j][i]=a[i][j];
}
}
printf("\nThe transpose of above matrix is:-\n");
for(i=0;i<c;++i)
{
for(j=0;j<r;++j)
{
printf("%d ",t[i][j]);
if(j==r-1)
printf("\n");
}

} }
void data(int aa[][10],int bb[][10],int r1,int c1,int r2,int c2)
{
printf("\nEnter elements of 1st matrix:- \n");
for(int i=0;i<r1;++i)
{
for(int j=0;j<c1;++j)
{
printf("\nEnter element a[%d][%d]:- ",i+1,j+1);
scanf("%d",&aa[i][j]);
}
}
printf("\nEnter elements of 2nd matrix:- \n");
for(int i=0;i<r2;++i)
{
for(int j=0;j<c2;++j)
{
printf("\nEnter element b[%d][%d]:- ",i+1,j+1);
scanf("%d",&bb[i][j]);
}
```

```
}
}
void multiply(int aa[][10],int bb[][10],int mult[][10],int r1,int c1,int r2,int c2)
{
for(int i=0;i<r1;++i)
{
for(int j=0;j<c2;++j)
{
mult[i][j]=0;
}
}
for(int i=0;i<r1;++i)
{
for(int j=0;j<c2;++j)
{
for(int k=0;k<c1;++k)
{
mult[i][j]+=aa[i][k]*bb[k][j];
}
}
}

}
void display(int mult[][10],int r1,int c2)
{
printf("\nProduct of the given matrices is:- \n");
for(int i=0;i<r1;++i)
{
for(int j=0;j<c2;++j)
{
printf("%d ",mult[i][j]);
if(j == c2-1)
printf("\n");
}
}
}
void sparse()
{
printf("\nEnter the number of rows and columns of matrix:-\n");
scanf("%d %d",&r,&c);
printf("\nEnter the elements of matrix:-\n");
for(i=0;i<r;++i)
{
for(j=0;j<c;++j)
{
scanf("%d",&a[i][j]);
```

```
if(a[i][j]==0)
{
++s;
}
}
}
if(s> (r*c)/2)
{
printf("\nThe given matrix is a sparse matrix as out of %d elements there are %d
zeros present in this matrix.\n",i*j,s);
}
else
printf("\nThe given matrix is not a sparse matrix.\nBecause out of %d elements, there
are
only %d Zeros present.\n",i*j,s);
}
```

OUTPUT:



```
Subtraction of the matrices is:
-4   2

6   5



This is a menu driven program for matrix.
**********************************************************
Enter 1 to add two matrices.
Enter 2 to transpose a matrix.
Enter 3 to multiply two matrices.
Enter 4 to subtract two matrices.
Enter 5 to determine wheather the given matrix is sparse or not.
Enter 6 to exit the program.
**********************************************************

Enter any of above options:- 5

Enter the number of rows and columns of matrix:-
2
3

Enter the elements of matrix:-
1
0
5
0
0
0

The given matrix is a sparse matrix as out of 6 elements there are 4 zeros present in this matrix.


This is a menu driven program for matrix.
**********************************************************
Enter 1 to add two matrices.
Enter 2 to transpose a matrix.
Enter 3 to multiply two matrices.
Enter 4 to subtract two matrices.
Enter 5 to determine wheather the given matrix is sparse or not.
Enter 6 to exit the program.
**********************************************************

Enter any of above options:-
```

```
Enter 5 to determine wheather the given matrix is sparse or not.
Enter 6 to exit the program.
**********************************************************

Enter any of above options:- 2

Enter the number of rows:- 2

Enter the number of columns:- 3

Enter the elements of matrix:-

Enter element a[1][1]:-1

Enter element a[1][2]:-2

Enter element a[1][3]:-6

Enter element a[2][1]:-4

Enter element a[2][2]:-3

Enter element a[2][3]:-8

The given matrix is:-
1  2  6
4  3  8

The transpose of above matrix is:-
1   4
2   3
6   8


This is a menu driven program for matrix.
**********************************************************
Enter 1 to add two matrices.
Enter 2 to transpose a matrix.
Enter 3 to multiply two matrices.
Enter 4 to subtract two matrices.
Enter 5 to determine wheather the given matrix is sparse or not.
Enter 6 to exit the program.
**********************************************************

Enter any of above options:-
```

```
Enter 2 to transpose a matrix.
Enter 3 to multiply two matrices.
Enter 4 to subtract two matrices.
Enter 5 to determine wheather the given matrix is sparse or not.
Enter 6 to exit the program.
*********************************************************

Enter any of above options:- 4

Enter the no. of rows of matrix:- 2

Enter the no. of columns:- 2

Enter elements of 1st matrix:-

Enter element a[1][1]:- 5

Enter element a[1][2]:- 6

Enter element a[2][1]:- 7

Enter element a[2][2]:- 8

Enter elements of 2nd matrix:-

Enter element b[1][1]:- 9

Enter element b[1][2]:- 4

Enter element b[2][1]:- 1

Enter element b[2][2]:- 3

Subtraction of the matrices is:
-4   2

6   5



This is a menu driven program for matrix.
*********************************************************
Enter 1 to add two matrices.
Enter 2 to transpose a matrix.
Enter 3 to multiply two matrices.
```

```
Enter 6 to exit the program.
*********************************************************

Enter any of above options:- 3

Enter no. of rows and columns of first matrix:-
2
3

Enter no. of rows and columns of second matrix:-
3
2

Enter elements  of 1st matrix:-

Enter element a[1][1]:- 1

Enter element a[1][2]:- 2

Enter element a[1][3]:- 3

Enter element a[2][1]:- 2

Enter element a[2][2]:- 5

Enter element a[2][3]:- 3

Enter elements of 2nd matrix:-

Enter element b[1][1]:- 2

Enter element b[1][2]:- 7

Enter element b[2][1]:- 6

Enter element b[2][2]:- 4

Enter element b[3][1]:- 3

Enter element b[3][2]:- 7

Product of the given matrices is:-
23  36
43  55
```

PRACTICAL 4

Design, Develop and Implement a menu driven Program in C for the following operations on RECURSION

a. Find factorial of an element
b. Find Fibonacci series
c. Find power of a number
d. Find sum of all digits
e. Exit

Support the program with appropriate functions for each of the above operations.

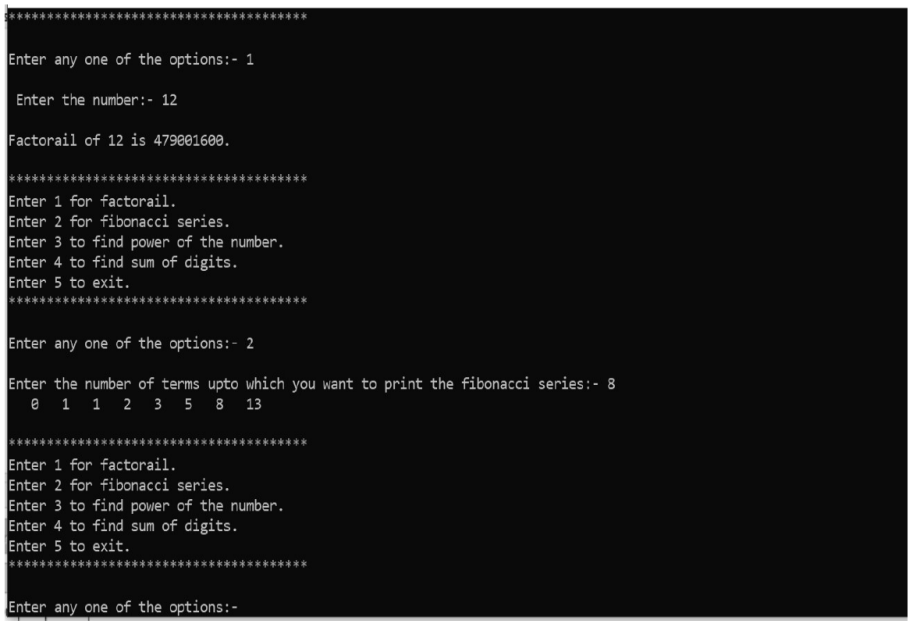
PROGRAM: -

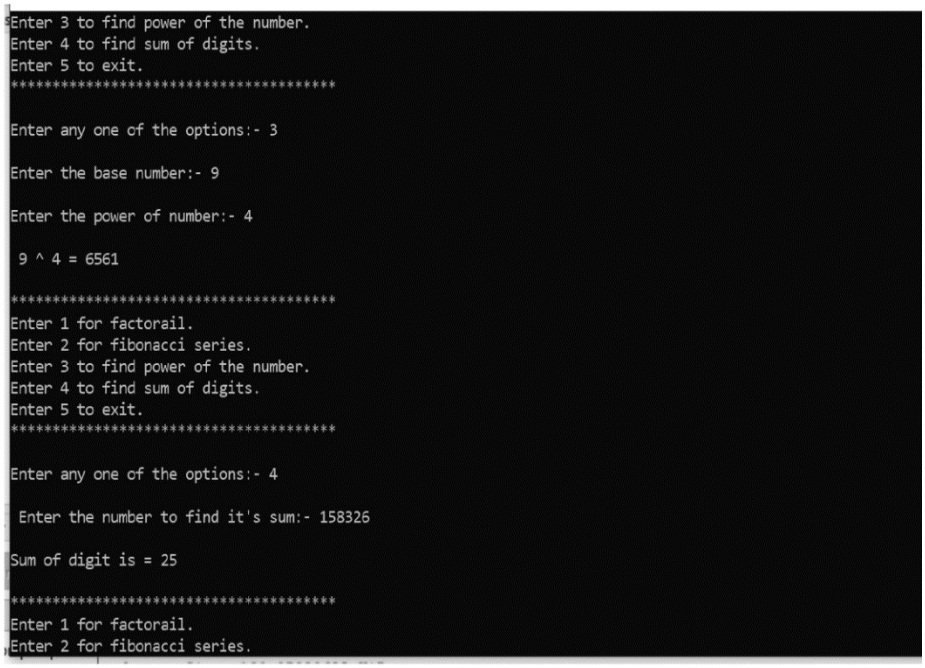
```
#include<stdio.h>
#include<stdlib.h>
int factorial(int);int fibonnaci(int);int power(int,int);int sum(int);
int main()
{
int ch,num,terms,i,fact,base,pow,result;
printf("This is a menu driven program based on recurssion.\n");
printf("**************************************************");
do
{
printf("\n\n****************************************\n");
printf("Enter 1 for factorail.\n");
printf("Enter 2 for fibonacci series.\n");
printf("Enter 3 to find power of the number.\n");
printf("Enter 4 to find sum of digits.\n");
printf("Enter 5 to exit.\n");
printf("****************************************\n\n");
printf("Enter any one of the options:- ");
scanf("%d",&ch);
switch(ch)
{
case 1:
printf("\n Enter the number:- ");
scanf("%d",&num);
fact=factorial(num);
printf("\nFactorail of %d is %d.",num,fact);
break;
case 2:
printf("\nEnter the number of terms upto which you want to print the
fibonacci series:- ");
```

```
scanf("%d",&terms);
for(i=0;i<terms;i++)
{
printf("\n %d",fibonnaci(i));
}
break;
case 3:
printf("\nEnter the base number:- ");
scanf("%d",&base);
printf("\nEnter the power of number:- ");
scanf("%d",&pow);
result=power(base,pow);
printf("\n %d ^ %d = %d",base,pow,result);
break;
case 4:
printf("\n Enter the number to find it's sum:- ");
scanf("%d",&num);
result=sum(num);
printf("\nSum of digit is = %d",result);
break;
case 5:
printf("\n******* THANKS TO USE *******\n");
exit(1);
default:
printf("\nOOPS, YOU HAVE ENTERED AN INVALID CHOICE.\n");
break;
}
}while(ch!=5);
return 0;
}// END OF MAIN
int factorial(int n)
{
if(n==0)
return(1);
return(n*factorial(n-1));
}
int fibonnaci(int num)
{
if(num==0||num==1)
{
return num;
}
else
{ return fibonnaci(num-1)+fibonnaci(num-2);
}
}
```

```
int power(int base,int pow)
{
if(pow!=0)
return(base*power(base,pow-1));
else
return 1;
}
int sum(int num)
{
if(num==0)
return 0;
return((num%10)+sum(num/10));
}
```

OUTPUT:





PRACTICAL 5

Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Branch, Sem, PhNo.

a. Create a SLL of N Students Data by using front insertion.
b. Display the status of SLL and count the number of nodes in it
c. Perform Insertion and Deletion at End of SLL
d. Perform Insertion and Deletion at Front of SLL
e. Perform Insertion and Deletion at given position of SLL
f. Reverse the elements of SLL
g. Exit

PROGRAM: -

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
struct student
{
int data;
struct student *next;
}s;
struct student *start = NULL;
struct student *create(struct student *);
int main()
{
start = create(start);
getch();
return 0;
}
struct student *create(struct student *start)
{
struct student *new_node, *ptr;
int num;
struct name
{
char fname[25];
char lname[25];
char sem[10];
char contact[12];
}d;
while(num!=-1)
{
printf("\nDo you want to enter next data:- ");
```

```
scanf("%d",&num);
new_node = (struct student*)malloc(sizeof(struct student));
new_node->data=num;
if(start==NULL)

{
new_node->next=NULL;
start=new_node;
}
else
{
ptr=start;
while(ptr->next!=NULL)
ptr=ptr->next;
ptr->next=new_node;
new_node->next=NULL;
}
printf("\nEnter student's details.\n");
printf("\nEnter your name: ");
printf("\nEnter your first name:- ");
scanf("%s",d.fname);
printf("\nEnter your last name:- ");
scanf("%s",d.lname);
printf("\nEnter the semester:- ");
scanf("%s",d.sem);
printf("\nEnter the contact number:- ");
scanf("%s",d.contact);
}
}
```

OUTPUT:

```
****************************************************************

The created list of students is as follow.
Student 1
Name:- harsh
Roll Number:- 17
Semester:- 2
Contact Number:- 9123250051

The created list of students is as follow.
Student 2
Name:- ravi
Roll Number:- 12
Semester:- 3
Contact Number:- 1234567890
```

PRACTICAL 6

Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)

a. Push an Element on to Stack
b. Pop an Element from Stack
c. Demonstrate how Stack can be used to check Palindrome
d. Demonstrate Overflow and Underflow situations on Stack
e. Display the status of Stack
f. Exit
Support the program with appropriate functions for each of the above operations

PROGRAM: -

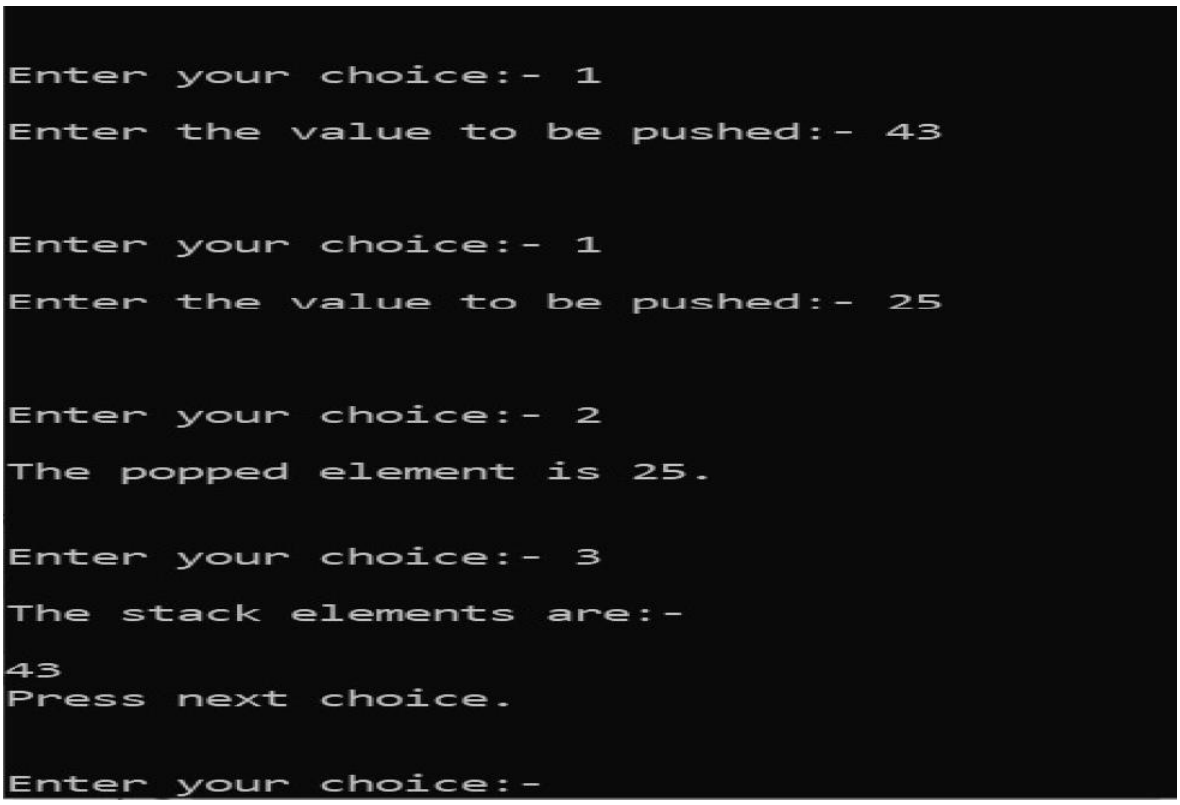
```
#include<stdio.h>
#include<process.h>
#include<string.h>
#define MAX 5
void push( ); void pop( ); void display( );void status( );void pus(char);char po( );void
str( );
int stack[MAX],choice,n,top,x,i;
char stac[50];
int to= -1;
char string[30];
int count=0,len;
int main( )
{
top= -1;
printf("\nSTACK operation using Array");
printf("\n----------------------------");
printf("\n1.PUSH\n2.POP\n3.DISPLAY\n4.STATUS\n5.CHECK
PALINDROME\n6.EXIT\n*****************************\n---------------------------
");
do
{
printf("\n\n\nEnter your choice:- ");
scanf("%d",&choice);
switch(choice)
{
case 1:
push();
break;
case 2:
```

```
pop();
break;
case 3:
display();
break;
case 4:
status();

break;
case 5:
str();
break;
case 6:
printf("\nExit point.\n");
break;
default:
printf("\nPlease enter VALID choice(1/2/3/4/5/6).");
}
}
while(choice!=6);
return 0;
}
void push()
{
if(top==MAX-1)
{
printf("\n\tStack is Overflow.");
}
else
{
top=top+1;
printf("\nEnter the value to be pushed:- ");
scanf("%d",&n);
stack[top]=n;
}
}
void pop()
{
if(top<0)
{
printf("\nStack is Underflow.");
}
else
{
printf("\nThe popped element is %d.",stack[top]);
top--;
```

```
}
}

void display()
{
if(top>=0)
{
printf("\nThe stack elements are:- \n");
for(i=top;i>=0;i--)
printf("\n%d",stack[i]);
printf("\nPress next choice.");
}
else
{
printf("\nStack is Empty.");
}
}
void status()
{
printf("\nYou have entered %d elements in this stack.\n",top+1);
if((4-top)==0)
{
printf("\nThe stack is full.\nYou can't insert anymore elements in it.\n");
}
else
printf("The number of more elements which can be inserted in this stack is %d.\n",4-
top);
}
void str()
{
printf("\nEnter the string:- ");
scanf("%s",string);
len=strlen(string);
for(i=0;i<len;i++)
{
pus(string[i]);
}
for(i=0;i<len;i++)
{
if(string[i]==po())
{
count++;
}
}

if(len==count)
```

```
printf("\nString is a palindrome.",string);
else
printf("\nString is not a palindrome.",string);
}
void pus(char ch)
{
to=to+1;
stac[to]=ch;
}
char po()
{
char c;
c=stac[to];
to=to-1;
return c;
}
```

OUTPUT:



PRACTICAL 7

Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers using Linked List

a. Push an Element on to Stack
b. Pop an Element from Stack
c. Demonstrate Overflow and Underflow situations on Stack
d. Display the status of Stack
e. Exit
Support the program with appropriate functions for each of the above operations

PROGRAM: -

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct Node
{
int data;
struct Node *next;
}*top = NULL;
void push(int);
void pop();
void display();
int main()
{
int choice, value;
printf("\n****** MENU ******\n");
printf("1. Push\n2. Pop\n3. Display\n4. Exit\n");
printf("******************\n");
while(1)
{
printf("\nEnter your choice: ");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("Enter the value to be insert: ");
scanf("%d", &value);
push(value);
break;
case 2:
pop();
break;
case 3:
display();
```

```
break;

case 4:
exit(0);
default:
printf("\nWrong selection!!! Please try again!!!\n");
}
}
}
void push(int value)
{
struct Node *newNode;
newNode = (struct Node*)malloc(sizeof(struct Node));
newNode->data = value;
if(top == NULL)
newNode->next = NULL;
else
newNode->next = top;
top = newNode;
printf("\nInsertion is Successfull.!!!\n");
}
void pop()
{
if(top == NULL)
printf("\nStack is Empty!!!\n");
else
{
struct Node *temp = top;
printf("\nDeleted element: %d", temp->data);
top = temp->next;
free(temp);
}
}
void display()
{
if(top == NULL)
printf("\nStack is Empty!!!\n");
else
{
struct Node *temp = top;
while(temp->next != NULL)
{
printf("%d-->",temp->data);
temp = temp -> next;

}
printf("%d-->NULL",temp->data);
}
}
```

OUTPUT:

```
****** MENU ******
1. Push
2. Pop
3. Display
4. Exit
******************

Enter your choice: 1
Enter the value to be insert: 32

Insertion is Successfull.!!!

Enter your choice: 1
Enter the value to be insert: 63

Insertion is Successfull.!!!

Enter your choice: 1
Enter the value to be insert: 64

Insertion is Successfull.!!!

Enter your choice: 2

Deleted element: 64
Enter your choice: 3
63-->32-->NULL
Enter your choice:
```

PRACTICAL 8

Design, Develop and Implement a menu driven Program in C for the following operations on Linear QUEUE of Characters (Array Implementation of Queue with maximum size MAX)

a. Insert an Element on to Linear QUEUE
b. Delete an Element from Linear QUEUE
c. Demonstrate Overflow and Underflow situations on Linear QUEUE
d. Display the status of Linear QUEUE
e. Exit
Support the program with appropriate functions for each of the above operations

PROGRAM: -

```
#include<stdio.h>
#include<process.h>
#define MAX 5
void insert();void del();void display();void status();
int arr[MAX];
int rear = - 1;
int front = - 1;
int main()
{
int ch;
printf("NOTE:- The maximum size of this queue is 5.\n");
printf("1.Insert(PUSH)\n2.Delete(POP)\n3.Display\n4.Current status\n5.EXIT \n\n");
while (1)
{
printf("\nPlease enter your choice : ");
scanf("%d", &ch);
switch (ch)
{
case 1:
insert();
break;
case 2:
del();
break;
case 3:
display();
break;
case 4:
```

```
status();
break;
case 5:
exit(1);
default:
printf("Please enter valid option.\n");
}

} }
void insert()
{
int add;
if (rear == MAX - 1)
printf("\nQueue OVERFLOW.\nYou cannot enter anymore elements in this queue.\n");
else
{
if (front == - 1)
front = 0;
printf("Enter element to insert:- ");
scanf("%d", &add);
rear = rear + 1;
arr[rear] = add;
}
}
void del()
{
if (front == - 1 || front > rear)
{
printf("\nQueue UNDERFLOW \n");
return ;
}
else
{
printf("Element deleted from queue is : %d\n",arr[front]);
front = front + 1;
}
}
void display()
{
int i;
if (front == - 1)
printf("\nQueue is empty \n");
else
{
printf("Queue is : \n");
for (i = front; i <= rear; i++)
```

```
printf("%d ",arr[i]);
printf("\n");
}
}
void status()
{
printf("\nYou have entered %d elements in this queue.\n",rear+1);
if((4-rear)==0)
{
printf("\nThe queue is full.\nYou can't insert anymore elements in it.\n");
}
else
printf("The number of more elements which can be inserted in this queue is %d.\n",4-
rear);
}
```

OUTPUT:

```
NOTE:- The maximum size of this queue is 5.
1.Insert(PUSH)
2.Delete(POP)
3.Display
4.Current status
5.EXIT


Please enter your choice : 1
Enter element to insert:-  23

Please enter your choice : 1
Enter element to insert:-  47

Please enter your choice : 1
Enter element to insert:-  78

Please enter your choice : 2
Element deleted from queue is : 23

Please enter your choice : 3
Queue is :
47 78

Please enter your choice : 4

You have entered 3 elements in this queue.
The number of more elements which can be inserted in this queue is 2.

Please enter your choice :
```

PRACTICAL 9

Design, Develop and Implement a menu driven Program in C for the following Searching and Sorting operations using Array

a. Perform Linear Search
b. Perform Binary Search
c. Demonstrate Selection Sorting operation
d. Demonstrate Bubble sorting operation
e. Demonstrate Insertion Sorting operation
f. Demonstrate Quick sorting operation
e. Exit
Support the program with appropriate functions for each of the above operations

PROGRAM: -

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
int a[20],n,s,i,j,min,temp,c=0,low,high,mid;
void linear(); void binary(); void selection(); void input(); void bubble(); void insertion();
void
quicksort(int [],int,int);
int main()
{
int ch;
printf("\n\nThis is a menu driven program for sorting of array.\n");
while(1)
{
printf("\n************************************************************\n");
printf("Enter 1 for performing linear search.\n");
printf("Enter 2 for performing binary search.\n");
printf("Enter 3 to sort array using selection sorting.\n");
printf("Enter 4 to sort array using bubble sorting.\n");
printf("Enter 5 to sort array using insertion sorting.\n");
printf("Enter 6 to sort array using quick sorting.\n");
printf("Enter 7 to exit program.\n");
printf("************************************************************\n\n");
printf("Enter any of above options:- ");
scanf("%d",&ch);
switch(ch)
{
case 1:
```

```
input();
linear();
break;
case 2:
input();
binary();
break;
case 3:
input();
selection();
break;
case 4:
input();
bubble();
break;
case 5:
input();
insertion();
break;
case 6:
input();
quicksort(a, 0, n-1);
printf("\nArray after sorting in ascending order:-\n");
for(i=0;i<n;i++)
{
printf("%d, ",a[i]);
}
break;
case 7:
exit (0);
default :
printf("\n OOPS, you have entered an invalid choice.\n\n");
}
}
return 0;
}
void linear()
{
printf("Enter a number to search:- ");
scanf("%d", &s);
for (i = 0; i < n; i++)
{
if (a[i] == s) {
printf("%d is present at location %d.\n", s, i+1);
c++;
}
```

```
}
if (c == 0)

printf("%d isn't present in the array.\n", s);
else
printf("%d is present %d times in the array.\n", s, c);
}
void binary()
{
printf("Enter value to find:- ");
scanf("%d", &s);
low = 0;
high = n - 1;
mid = (low+high)/2;
while (low <= high)
{
if(a[mid] < s)
low = mid + 1;
else if (a[mid] == s)
{
printf("%d found at location %d.", s, mid+1);
break;
}
else
high = mid - 1;
mid = (low + high)/2;
}
if(low > high)
printf("Not found! %d isn't present in the list.", s);
}
void input()
{
printf("\nEnter no. of elements:- ");
scanf("%d",&n);
printf("\nEnter %d elements:- \n",n);
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
}
void selection()
{
for(i=0;i<n-1;i++)
{
min=i;
```

```
for(j=i+1;j<n;j++)
{
if(a[min]>a[j])
min=j;
}
if(min!=i)
{
temp=a[i];
a[i]=a[min];
a[min]=temp;
}
}
printf("\nThe sorted array in a ascending order is:- \n");
for(i=0;i<n;i++)
{
printf("%d, ",a[i]);
}
}
void bubble()
{
for(i=0;i<n-1;i++)
{
for(j=0;j<(n-i-1);j++)
{
if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
printf("\nArray elements in ascending order is:-\n");
for(i=0;i<n;i++)
{
printf("%d, ",a[i]);
}
}
void insertion()
{
for(i=1;i<=n-1;i++)
{

j=i;
while(j>0 && a[j]<a[j-1])
{
```

```
temp=a[j];
a[j]=a[j-1];
a[j-1]=temp;
j--;
}
}
printf("\nArray in ascending order is:- \n");
for(i=0;i<=n-1;i++)
{
printf("%d, ",a[i]);
}
}
void quicksort(int a[],int low,int high)
{
int p;
if(low<high)
{
p=low;
i=low;
j=high;
while(i<j)
{
while(a[i]<=a[p] && i<=high)
{
i++;
}
while(a[j]>a[p] && j>=low)
{
j--;
}
if(i<j)
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
temp=a[j];
a[j]=a[p];

a[p]=temp;
quicksort(a,low,j-1);
quicksort(a,j+1,high);
}
}
```

OUTPUT:

```
Enter any of above options:- 1

Enter no. of elements:- 3

Enter 3 elements:-
17
46
42
Enter a number to search:- 46
46 is present at location 2.
46 is present 1 times in the array.

************************************************************
Enter 1 for performing linear search.
Enter 2 for performing binary search.
Enter 3 to sort array using selection sorting.
Enter 4 to sort array using bubble sorting.
Enter 5 to sort array using insertion sorting.
Enter 6 to sort array using quick sorting.
Enter 7 to exit program.
************************************************************

Enter any of above options:- 2

Enter no. of elements:- 3

Enter 3 elements:-
12
23
34
Enter value to find:- 23
23 found at location 2.
************************************************************
Enter 1 for performing linear search.
Enter 2 for performing binary search.
Enter 3 to sort array using selection sorting.
Enter 4 to sort array using bubble sorting.
Enter 5 to sort array using insertion sorting.
Enter 6 to sort array using quick sorting.
Enter 7 to exit program.
```

```
Enter 3 to sort array using selection sorting.
Enter 4 to sort array using bubble sorting.
Enter 5 to sort array using insertion sorting.
Enter 6 to sort array using quick sorting.
Enter 7 to exit program.
************************************************************

Enter any of above options:- 3

Enter no. of elements:- 3

Enter 3 elements:-
27
64
29

The sorted array in a ascending order is:-
27, 29, 64,
************************************************************
Enter 1 for performing linear search.
Enter 2 for performing binary search.
Enter 3 to sort array using selection sorting.
Enter 4 to sort array using bubble sorting.
Enter 5 to sort array using insertion sorting.
Enter 6 to sort array using quick sorting.
Enter 7 to exit program.
************************************************************

Enter any of above options:- 4

Enter no. of elements:- 3

Enter 3 elements:-
15
37
32

Array elements in ascending order is:-
15,  32,  37,
************************************************************
Enter 1 for performing linear search.
```

```
Enter no. of elements:- 3

Enter 3 elements:-
16
32
86

Array in ascending order is:-
16,  32,  86,
************************************************************
Enter 1 for performing linear search.
Enter 2 for performing binary search.
Enter 3 to sort array using selection sorting.
Enter 4 to sort array using bubble sorting.
Enter 5 to sort array using insertion sorting.
Enter 6 to sort array using quick sorting.
Enter 7 to exit program.
************************************************************

Enter any of above options:- 6

Enter no. of elements:- 3

Enter 3 elements:-
18
65
35

Array after sorting in ascending order:-
18,  35,  65,
************************************************************
Enter 1 for performing linear search.
Enter 2 for performing binary search.
Enter 3 to sort array using selection sorting.
Enter 4 to sort array using bubble sorting.
Enter 5 to sort array using insertion sorting.
Enter 6 to sort array using quick sorting.
Enter 7 to exit program.
************************************************************

Enter any of above options:-
```