Date: 12/03/2022

EXPERIMENT NO. 5

Q. To study and demonstrate 1D AND 2D array.

PROGRAM STATEMENTS:

a. Write a C program to implement Linear and Binary Search using array and also display position of entered number in a given list.

b. Write a C program to find the transpose of entered matrix.

c. Write a C program to implement Insertion Sort and Selection sort.

THEORY:

1-D ARRAY:

Syntax: data\_type array\_name[array\_size];

2-D ARRAY:

Syntax: data\_type array\_name[rows][columns];

We used concept of array and loops to solve the problems given in the activity.

Program a)

Linear search:

#include <stdio.h>

int main()

{

int a[100],i,x,n,c=0;

printf("Enter the size of array: ");

scanf("%d",&n);

printf("Enter the element to be searched: ");

scanf("%d",&x);

printf("Enter the elements of the array: ");

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

scanf("%d\n",&a[i]);

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

{

if(a[i]==x)

{

c++;

printf("Element %d is found at index %d",x,i);

}

}

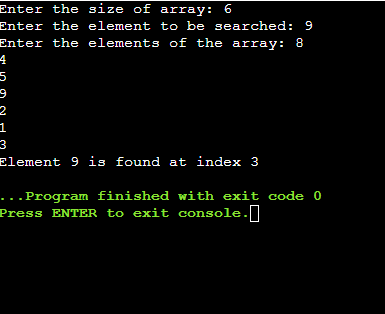
if(c==0)

printf("Element not found");

return 0;

}

OUTPUT:



Binary Search:

#include <stdio.h>

int main()

{

int a[100],i,x,n,first,last,middle;

printf("Enter the size of array: ");

scanf("%d",&n);

printf("Enter the element to be searched: ");

scanf("%d",&x);

printf("Enter the elements of the array: ");

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

scanf("%d\n",&a[i]);

first=0;

last=n-1;

middle=(first+last)/2;

while(first<=last)

{

if(a[middle]<x)

{

first=middle+1;

}

else if(a[middle]==x)

{

printf("Element %d is found at index %d\n",x,middle);

break;

}

else

{

last=middle+1;

}

middle=(first+last)/2;

}

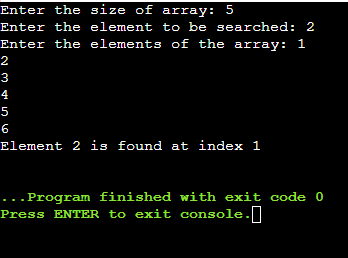
if(first>last)

printf("Element not found");

return 0;

}

OUTPUT:



Program b)

#include <stdio.h>

int main()

{

int a[100][100],trans[100][100],i,j,r,c;

printf("Enter the rows and columns: ");

scanf("%d %d",&r,&c);

printf("Enter the elements of the matrix:\n");

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

printf("Enter the element a%d%d: ",i+1,j+1);

scanf("%d",&a[i][j]);

}

}

printf("Entered matrix:\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++)

{

trans[j][i]=a[i][j];

}

}

printf("Transpose of entered matrix:\n");

for(i=0;i<c;i++)

{

for(j=0;j<r;j++)

{

printf("%d",trans[i][j]);

if(j==r-1)

printf("\n");

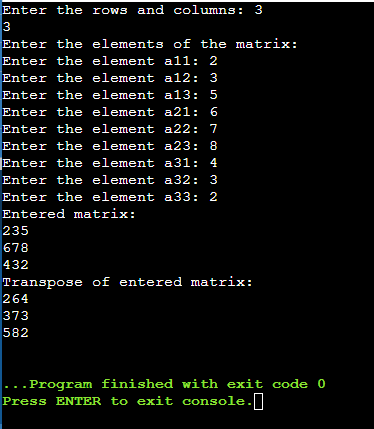
}

}

return 0;

}

OUTPUT:



Program c)

Selection Sort:

#include <stdio.h>

int main()

{

int a[100],n,temp,i,j;

printf("Enter the size of array: ");

scanf("%d",&n);

printf("Enter the elements of the array: ");

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

scanf("%d",&a[i]);

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

{

for(j=i+1;j<n;j++)

{

if(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf("Sorted array:\n");

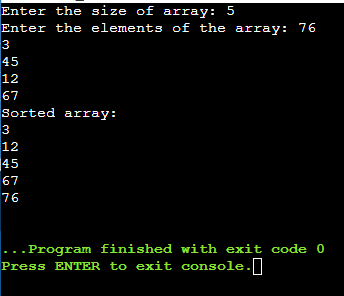
for(i=0;i<n;i++)

printf("%d\n",a[i]);

return 0;

}

OUTPUT:



Insertion Sort:

#include <stdio.h>

int main()

{

int a[100],n,temp,i,j;

printf("Enter the size of array: ");

scanf("%d",&n);

printf("Enter the elements of the array: ");

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

scanf("%d",&a[i]);

for(i=0;i<n-1;i++)

{

j=i+1;

temp=a[j];

while(j>0 && a[j-1]>temp)

{

a[j]=a[j-1];

j--;

}

a[j]=temp;

}

printf("Sorted array:\n");

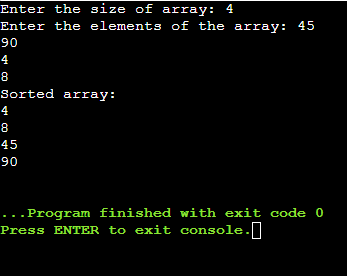
for(i=0;i<n;i++)

printf("%d\n",a[i]);

return 0;

}

OUTPUT:



CONCLUSION: We learned about the use of 2D array in solving matrix related problems. We learned how to arrange elements of array in ascending order.