# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 1

1) To Implement One Dimensional and Two Dimension Array programs using C.

Theory: 1. WAP in C to store and print an array.

WAP in C to print array in a reverse order.

WAP in C to find Min and Max elements of an array.

WAP in C to find sum of two matrices

Algorithm (1) to store and print an array:

BEGIN

INT i,n

READ No Of Elements=N INIT ARRAY[N]

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

READ Element And Store In ARRAY[i]

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

PRINT(ARRAY[i])

END

CODE:

#include <stdio.h>

#include <stdlib.h>

int main()

{ int n,i;

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

scanf("%d",&n);

int a[n];

printf("\nEnter the elements of the array:\n");

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

{

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

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

}

printf("Entered elements array:\n");

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

{

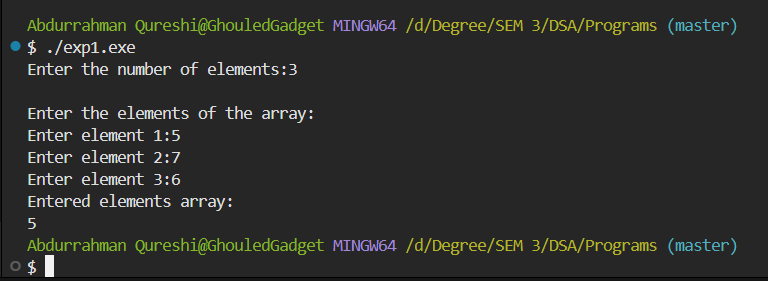
printf("%d ",a[i]);

return 0;

}

}

Output:



Algorithm (2) to print array in a reverse order:

BEGIN

INT i,n

READ No Of Elements=N

INIT ARRAY[N]

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

READ element and store in ARRAY[i]

FOR(i=n-1;i>=0;i--)

PRINT(ARRAY[i])

END

CODE:

#include <stdio.h>

#include <stdlib.h>

int main() {

int n,i;

int a[10];

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

scanf("%d",&n);

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

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

{

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

}

printf("Reverse of the array:\n");

for(i=n-1; i>=0; i--)

{

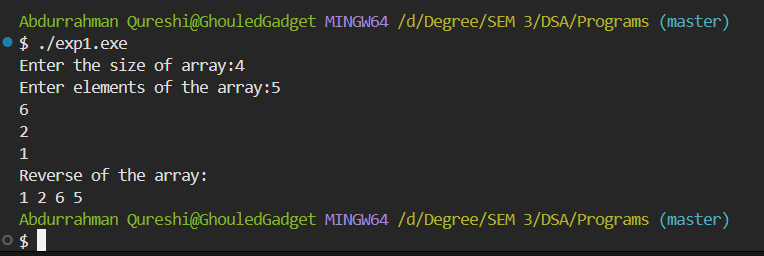
printf("%d ",a[i]);

}

return 0;

}

Output:



Algorithm (3) to to find Min and Max elements of an array:

BEGIN

INT i,n,min,max

READ NO OF ELEMENTS=N

INIT ARRAY1[N]

min=array1[0];

max=array1[0];

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

READ ELEMENT AND STORE IN ARRAY1[i]

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

if (array1[i]<min)

min=array1[i]

if (array1[i]>max)

max=array1[i]

PRINT(MAX)

PRINT(MIN)

END

CODE:

#include <stdio.h>

#include <stdlib.h>

int main() {

int a[20], n,i,j,c;

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

scanf("%d",&n);

printf("Enter the numbers:");

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])

{

c=a[i];

a[i]=a[j];

a[j]=c;

}

}

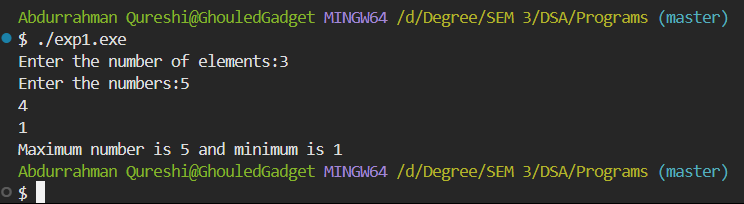
}

printf("Maximum number is %d and minimum is %d",a[n-1],a[0]);

return 0;

}

Output:



Algorithm (4) to find sum of two matrices :

BEGIN

READ matrix 1 and matrix 2 rows and column

Init a[r][c],b[r][c],d[r][c]

for i=1 to rows[a]

for j=1 to columns [a]

READ matrix 1 values

for i=1 to rows[b]

for j=1 to columns [b]

READ matrix 2 values

for i=1 to rows[d]

for j=1 to columns [d]

d[i,j]= a[i,j]+ b[i,j];

PRINT d[i,j];

END

CODE:

#include <stdio.h>

#include <stdlib.h>

int main()

{

int r,c,i,j;

printf("Enter numbers of rows and columns:");

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

int a[r][c];

int b[r][c];

int d[r][c];

printf("\nEnter elements of matrix 1\n");

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

{

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

{

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

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

}

}

printf("\nEnter elements of matrix 2\n");

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

{

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

{

printf("Enter b%d%d:",i+1,j+1);

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

}

}

printf("\nMatrix 1\n");

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

{

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

{

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

}

printf("\n");

}

printf("\nMatrix 2\n");

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

{

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

{

printf("%d\t",b[i][j]);

}

printf("\n");

}

printf("\nSum of the matrices:\n");

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

{

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

{

d[i][j]=a[i][j]+b[i][j];

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

}

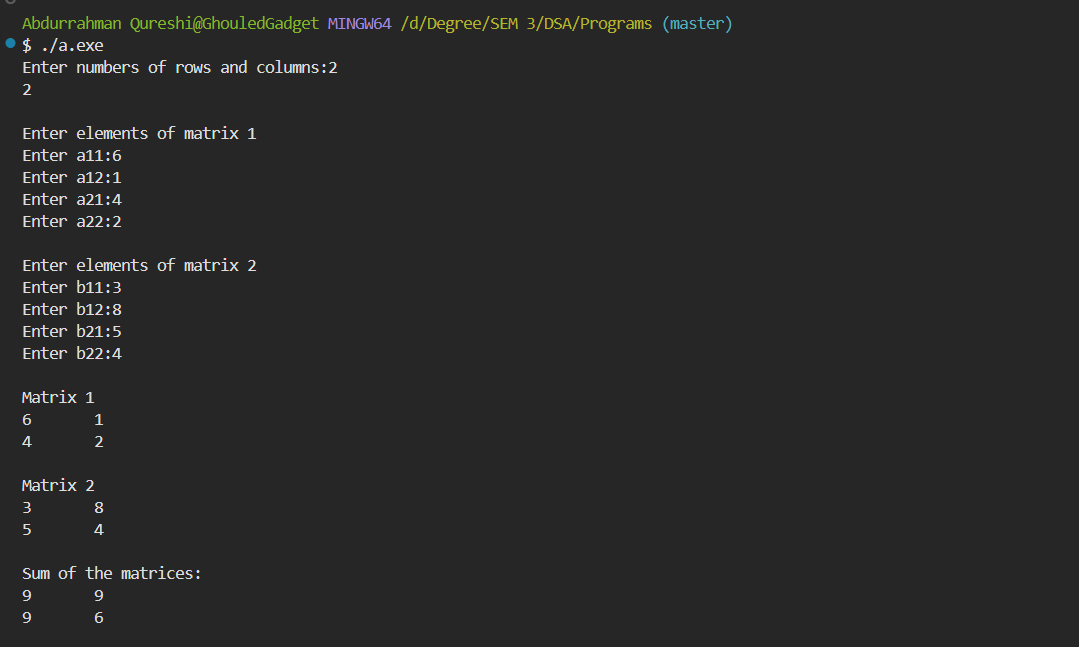
printf("\n");

}

return 0;

}

Output:



Tools used :

Software: Dev c++

Hardware: Lab Computers

**References:** No references used.

Conclusion:

I have understood and used the basic concepts and principles of various linked lists, stacks and queues. Stack is a container of objects that are inserted and removed according to the last-in first-out (LIFO) principle.

A one-dimensional array stores a single list of various elements having a similar data type.

A two-dimensional array stores an array of various arrays, a list of various lists, or an array of various one-dimensional arrays.

We can initialize each type of array at the time of declaration and could access any of their elements after that.