//Q1:WAP to compare two strings

#include <stdio.h>

#include<string.h>

int main()

{ int i=0,f=0,l1,l2;

char str1[100],str2[100];

printf("input a first string:");

fgets(str1,100,stdin);

printf("input a second string:");

fgets(str2,100,stdin);

l1=strlen(str1);

l2=strlen(str2);

if(l1!=l2)

f=1;

else{

while(str1[i]!='\0'){

while(str1[i]!=str2[i])

{

f=1;

break;

}

i++;

}

}

if(f==1)

printf("strings are not same");

else

printf("Both strings are same");

return 0;

}

**Output:**

**input a first string: mohammad faiz**

**input a second string: mohammad faiz**

**Both strings are same**

**input a first string:mohmmad faiz**

**input a second string:vo ivek singh**

**strings are not same**

**//Q2:Define code to pass an array to find the largest and smallest elements of an array**

#include <stdio.h>

void maxMin(int[],int);

int main()

{

int arr[20],n,i;

printf("input a number of an elements");

scanf("%d",&n);

printf("input an elements");

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

{

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

}

maxMin(arr,n);

return 0;

}

void maxMin(int arr[],int n)

{

int min,max,i;

min=arr[0];

max=arr[0];

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

{

if(max<arr[i]){

max=arr[i];

}

if(min>arr[i]){

min=arr[i];

}

}

printf("max=%d MIn=%d", max,min);

}

**OUTPUT**

**input a number of an elements5**

**input an elements1 2 3 4 5**

**max=5 MIn=1**

**//Q3: Fnd the equilibrium Index of an array:**

#include <stdio.h>

int findEquilibriumIndex(int arr[], int size) {

int totalSum = 0;

int leftSum = 0;

for (int i = 0; i < size; i++) {

totalSum += arr[i];

}

for (int i = 0; i < size; i++) {

totalSum -= arr[i];

if (leftSum == totalSum) {

return i;

}

// Update the left sum

leftSum += arr[i];

}

return -1;

}

int main() {

int size;

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

scanf("%d", &size);­

int arr[size];

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

for (int i = 0; i < size; i++) {

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

}

int equilibriumIndex = findEquilibriumIndex(arr, size);

if (equilibriumIndex != -1) {

printf("The equilibrium index is %d\n", equilibriumIndex);

} else {

printf("No equilibrium index found.\n");

}

return 0;

}

**OUTPUT**:

Enter the size of the array: 7

Enter the elements of the array:

1 2 3 4 3 2 1

The equilibrium index is 3

//Q4:WAP to find the sum of odd position of an array.

#include<stdio.h>

int main()

{

int a[10],n,k,i,j,s=0,t;

printf("input a number less than 10\n");

scanf("%d", &n);

printf("input elements of an array\n");

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

{

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

}

for(i=1;i<=n;i=i+2)

{

s=s+a[i-1];

}

printf(" the sum of odd position is %d\n",s);

return 0;

}

**OUTPUT**

input a number less than 10

5

input elements of an array

1 2 3 4 5

the sum of odd position is 9

**//Q5:WAP to input matrix and print the count of all positive even numbers**.

#include<stdio.h>

int main()

{

int a[10][10],m,n,i,j,count=0;

printf("Input a no. of rows and column\n");

scanf("%d%d",&n,&m);

printf("input elements of an array\n");

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

{

for(j=0;j<n;j++){

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

printf("\n");

}

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

{

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

{

if(a[i][j]>0&&a[i][j]%2==0)

{

count++;

}

}

}

printf(" The total psitive even number is is %d\n",count);

return 0;

}

**OUTPUT:**

Input a no. of rows and column

3 3

input elements of an array

1 2 3 4 5 6 6 7 9

The total psitive even number is is 4

**//Q6:WAP to input matrix and print the sum of each row of matrix.**

#include<stdio.h>

int main()

{

int a[10][10],m,n,i,j;

printf("Input a no. of rows and column\n");

scanf("%d%d",&n,&m);

printf("input elements of an array\n");

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

{

for(j=0;j<n;j++){

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

printf("\n");

}

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

{ int sum=0;

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

{

sum+=a[i][j];

}

printf("The sum of %d row is %d\n",i+1,sum);

}

return 0;

}

**OUTPUT:**

Input a no. of rows and column

3 3

input elements of an array

1 2 3

4 5 6

7 8 9

The sum of 1 row is 6

The sum of 2 row is 15

The sum of 3 row is 24

**//Q7:WAP to transpose of a matrix**

#include <stdio.h>

int main()

{

int m, n, c, d, matrix[10][10], transpose[10][10];

printf("Enter the number of rows and columns of matrix : ");

scanf("%d%d", &m, &n);

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

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

{

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

{

scanf("%d", &matrix[c][d]);

}

}

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

{

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

{

transpose[d][c] = matrix[c][d];

}

}

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

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

{

for (d = 0; d < m; d++)

{

printf("%d\t", transpose[c][d]);

}

printf("\n");

}

return 0;

}

**OUTPUT:**

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

Enter the elements of matrix :

1 2 3

4 5 6

7 8 9

Transpose of entered matrix :-

1 4 7

2 5 8

3 6 9

**//Q8: WAP to check a input matrix is sparse or not**:

#include <stdio.h>

int main()

{

static int array[10][10];

int i, j, m, n;

int counter = 0;

printf("Enter the order of the matix : ");

scanf("%d %d", &m, &n);

printf("Enter the co-efficients of the matix \n");

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

{

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

{

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

if (array[i][j] == 0)

{

counter++;

}

}

}

if (counter > ((m \* n) / 2))

{

printf("The given matrix is sparse matrix \n");

}

else

{

printf("The given matrix is not a sparse matrix \n");

}

printf("There are %d number of zeros", counter);

return 0;

}

**OUTPUT:**

Enter the order of the matix : 3 3

Enter the co-efficients of the matix

1 0 2 0 3 0 4 0 0

The given matrix is sparse matrix

There are 5 number of zeros

**//Q9: WAP to mutiplication of two matrix:**

#include <stdio.h>

int main()

{

int m, n, p, q, c, d, k, sum = 0;

int first[10][10], second[10][10], multiply[10][10];

printf("Enter the number of rows and columns of first matrix\n");

scanf("%d%d", &m, &n);

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

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

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

scanf("%d", &first[c][d]);

printf("Enter the number of rows and columns of second matrix\n");

scanf("%d%d", &p, &q);

if (n != p)

printf("Matrices with entered orders can't be multiplied with each other.\n");

else

{

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

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

for (d = 0; d < q; d++)

scanf("%d", &second[c][d]);

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

{

for (d = 0; d < q; d++)

{

for (k = 0; k < p; k++)

{

sum = sum + first[c][k] \* second[k][d];

}

multiply[c][d] = sum;

sum = 0;

}

}

printf("Product of entered matrices:-\n");

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

{

for (d = 0; d < q; d++)

printf("%d\t", multiply[c][d]);

printf("\n");

}

}

return 0;

}

**OUTPUT**

Enter the number of rows and columns of first matrix

3 3

Enter the elements of first matrix

1 2 3 4 5 6 7 8 9

Enter the number of rows and columns of second matrix

3 3

Enter the elements of second matrix

9 8 7 6 5 4 3 2 1

Product of entered matrices:-

30 24 18

84 69 54

138 114 90

**//Q10:WAP to check a given matrix is symmetric or not:**

#include<stdio.h>

#include <stdlib.h>

int main()

{

int i, j, a[10][10], b[10][10], n, temp;

printf("\nEnter the dimensions of matrix:: ");

scanf("%d", &n);

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

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

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

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

printf("\nOriginal Matrix is:: ");

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

{

printf("\n");

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

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

}

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

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

{

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

}

printf("\nTranspose Matrix is:: ");

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

{

printf("\n");

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

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

}

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

{

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

{

if (a[i][j] != b[i][j])

{

printf("\nMatrix is not symmetric :(\n");

exit(0);

}

}

}

printf("\nMatrix is symmetric :)\n");

return 0;

}

**OUTPUT:**

**Ex1:**

Enter the dimensions of matrix:: 3 3

Enter the elements of matrix::

1 7 3 7 4 -5 3 -5 6

Original Matrix is::

3 1 7

3 7 4

-5 3 -5

Transpose Matrix is::

3 3 -5

1 7 3

7 4 -5

Matrix is not symmetric :

**Ex2:**

Enter the dimensions of matrix:: 3 3

Enter the elements of matrix::

3 1 1 1 1 1 1 1 1

Original Matrix is::

3 1 1

1 1 1

1 1 1

Transpose Matrix is::

3 1 1

1 1 1

1 1 1

Matrix is symmetric :)