* Problem 1

**Sum of 2 Matrices**

**AIM:-** To write a ‘c’ program to take input of 2 Matrices and display the sum of it using **Array.**

**THEORY:-** The collection of multiple Data in a single variable is known as **Array**. For example it is written as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start.

Step 2:- Declare a[2][2], b[2][2], c[2][2] for Array and i, j for loops.

Step 3:- Check FOR condition i<?, if it is true go to step 4 otherwise go to step 8.

Step 4:- Check FOR condition j<?, if it is true go to step 5 otherwise go to step 7.

Step 5:- Read variable ‘a’ by nested loop.

Step 6:- Increment j+1 and go to step 4.

Step 7:- Increment i+1 and go to step 3.

Step 8:- Check FOR condition i<?, if it is true go to step 9 otherwise go to step 13.

Step 9:- Check FOR condition j<?, if it is true go to step 10 otherwise go to step 12.

Step 10:- Read variable ‘b’ by nested loop.

Step 11:- Increment j+1 and go to step 9.

Step 12:- Increment i+1 and go to step 8.

Step 13:- Check FOR condition i<?, if it is true go to step 14 otherwise go to step 19.

Step 14:- Check FOR condition j<?, if it is true go to step 15 otherwise go to step 18.

Step 15:- Calculate Sum by c[i][j]=a[i][j]+b[i][j]; and values are stored in variable ‘c’.

Step 16:- Print Values of ‘c’ variable.

Step 17:- Increment j+1 and go to step 14.

Step 18:- Increment i+1 and go to step 13.

Step 19:- Stop.

**Program:-**

#include<stdio.h>

int main()

{

int a[2][2],b[2][2],c[2][2];

int i,j;

printf("Enter the Values in the 1st Matrix!!\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Row=%d And Column=%d\n",i,j);

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

}

}

printf("\n\nEnter the Values in the 2nd Matrix!!\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("Row=%d And Column=%d\n",i,j);

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

}

}

printf("\n\nThe Sum of 2 Matrices !!\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

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

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

}

printf("\n");

}

return 0;

}

**Output:-**

Enter the Values in the 1st Matrix!!

Row=0 And Column=0

1

Row=0 And Column=1

2

Row=1 And Column=0

1

Row=1 And Column=1

2

Enter the Values in the 2nd Matrix!!

Row=0 And Column=0

1

Row=0 And Column=1

2

Row=1 And Column=0

1

Row=1 And Column=1

2

The Sum of 2 Matrices !!

24

24

**Observation:-** After performing the experiment we observed that Values can be stored in a Multi-dimensional Array and we were able to find out the Sum of 2 Matrices. The Sum was stored in ‘c’ variable. It took 0.31 sec Compilation Time.

* **Problem 2**

**Temperature Conversion**

**AIM:-** To write a ‘c’ program to take input of Temperature in Celsius and display the Temperature in Fahrenheit in one Multi-dimensional **Array.**

**THEORY:-** The collection of multiple Data in a single variable is known as **Array**. For example it is written as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start.

Step 2:- Declare variables a[2][5] for Array and i, j for loops, ‘f’ for Fahrenheit.

Step 3:- Check FOR condition i<?, if it is true go to step 4 otherwise go to step 14.

Step 4:- Check Condition if (i==0), if it is true go to step 5 otherwise go to step 8.

Step 5:- Check FOR condition j<?, if it is true go to step 6 otherwise go to step 13.

Step 6:- Read variable ‘a’ by nested loop for 1st Row.

Step 7:- Increment j+1 and go to step 5.

Step 8:- else

Step 9:- Check FOR condition j<?, if it is true go to step 10 otherwise go to step 13.

Step 10:- Calculate Temperature by f=(a[0][j]\*9/5+32;.

Step 11:- Assign values to 2nd Row of ‘a’ variable by a[i][j]=f;.

Step 12:- Increment j+1 and go to step 9.

Step 13:- Increment i+1 and go to step 3.

Step 14:- Check FOR condition i<?, if it is true go to step 15 otherwise go to step 19.

Step 15:- Check FOR condition j<?, if it is true go to step 16 otherwise go to step 18.

Step 16:- Print variable ‘a[i][j]’.

Step 17:- Increment j+1 and go to step 15.

Step 18:- Increment i+1 and go to step 14.

Step 19:- Stop.

**Program:-**

#include<stdio.h>

int main()

{

int a[2][5];

int i,j,f;

printf("Enter the Temperature in Degrees!!\n");

for(i=0;i<2;i++)

{

if(i==0)

{

for(j=0;j<5;j++)

{

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

}

}

else

{

for(j=0;j<5;j++)

{

f=(a[0][j]\*9/5)+32;

a[i][j]=f;

}

}

}

printf("\n\nCelsius To Fahrenheit\n");

for(i=0;i<2;i++)

{

for(j=0;j<5;j++)

{

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

}

printf("\n");

}

return 0;

}

**Output:-**

Enter the Temperature in Degrees!!

28

89

-23

23

13

Celsius To Fahrenheit

28 89 -23 23 13

82 192 -9 73 55

**Observation :-** After performing the experiment we observed that Values can be stored in a Multi-dimensional Array and we were able to convert the temperature from Celsius to Fahrenheit in a given Matrix. The Result of Temperature Conversion was stored in ‘f’ variable and then passed to ‘a’ variable. It took 0.20 sec Compilation Time.

* **Problem 3**

**Transpose of Matrix**

**AIM:-** To write a ‘c’ program to take input from the user of a Matrix and display its Transpose in Multi-dimensional **Array.**

**THEORY:-** The collection of multiple Data in a single variable is known as **Array**. For example it is written as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start.

Step 2:- Declare variables a[3][3], b[3][3] for Array and i, j for loops.

Step 3:- Check FOR condition i<?, if it is true go to step 4 otherwise go to step 8.

Step 4:- Check FOR condition j<?, if it is true go to step 5 otherwise go to step 13.

Step 5:- Read variable ‘a’ by nested loop.

Step 6:- Increment j+1 and go to step 4.

Step 7:- Increment i+1 and go to step 3.

Step 8:- Check FOR condition i<?, if it is true go to step 9 otherwise go to step13 .

Step 9:- Check FOR condition j<?, if it is true go to step 10 otherwise go to step 12.

Step 10:- Print the values of ‘a’ in matrix form.

Step 11:- Increment j+1 and go to step 9.

Step 12:- Increment i+1 and go to step 8.

Step 14:- Check FOR condition i<?, if it is true go to step 15 otherwise go to step 19.

Step 15:- Check FOR condition j<?, if it is true go to step 16 otherwise go to step 18.

Step 16:- Assign b[i][j]=a[j][i], i.e rows of ‘a’ will become ‘b’ variable’s column and columns of ‘a’ will become ‘b’ variable’s row .

Step 17:- Print the Transposed Matrix.

Step 18:- Increment j+1 and go to step 15.

Step 19:- Increment i+1 and go to step 14.

Step 20:- Stop.

**Program:-**

#include<stdio.h>

int main()

{

int a[3][3],b[3][3];

int i,j;

printf("Enter the Values in the Matrix!!\n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

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

}

}

printf("\n\nThe Matrix\n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

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

}

printf("\n");

}

printf("\n\nThe Transpose of the Matrix !!\n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

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

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

}

printf("\n");

}

return 0;

}

**Output:-**

Enter the Values in the Matrix!!

12

34

32

45

65

36

16

41

12

The Matrix

12 34 32

45 65 36

16 41 12

The Transpose of the Matrix !!

12 45 16

34 65 41

32 36 12

**Observation :-** After performing the experiment we observed that Values can be stored in a Multi-dimensional Array and we were able to find the Transpose of the Matrix. By assigning rows with columns and columns with rows we got the Transposed Matrix . It took 0.16 sec Compilation Time.