ARRAYS

Ex no. 2 a	LARGEST ELEMENT
Date:	

AIM:

- To Find the largest element in an array.

Algorithm:

Step 1 : Start.

Step 2 : Declare required variables and get input from the user.

Step 3 : create a for loop which runs till the last element of the array and check if the number is largest .if yes, Store it in a separate variable else continue iteration.

Step 4 : Print the largest number in the array.

Step 5 : Stop.

Source code:

```
#include <stdio.h>
int main(){
   int i,n;
   float arr[100];
   printf("Enter total number of elements(1 to 100): ");
   scanf("%d",&n);
   printf("\n");
   for(i=0;i<n;++i)  /* Stores number entered by user. */
   {
     printf("Enter Number %d: ",i+1);
     scanf("%f",&arr[i]);
   }
```

```
    printf("Largest element = %.2f",arr[0]);

    return 0;

}
```

Sample output:

```
 Enter total number of elements(1 to 100): 8

Enter Number 1: 23.4

Enter Number 2: -34.5

Enter Number 3: 50

Enter Number 4: 33.5

Enter Number 5: 55.5

Enter Number 6: 43.7

Enter Number 7: 5.7

Enter Number 8: -66.5
```

Result;

The program is executed and is the result is found to be right.

Ex no. 1 b	MATRIX ADDITION
Date:	

AIM:

To find the resultant of addition of two matrix.

Algorithm:

Step 1 : Start

Step 2 : Declare required variables and get input from the user.

Step 3 : call the function add() which performs addition of matrix.

Step 4 : The add function contains a for loop which adds using the formula c[i][j]=a[i][j] + b[i][j]

Step 5 : Print the resultant matrix

Step 6 : Stop.

Source code:

```
#include <stdio.h>
int add()
{
  int m, n, c, d, first[10][10], second[10][10], sum[10][10];
  printf("Enter the number of rows and columns of 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 elements of second matrix\n");

  for (c = 0; c < m; c++)
    for (d = 0 ; d < n; d++)
        scanf("%d", &second[c][d]);
  printf("Sum of entered matrices:-\n");
   for (c = 0; c < m; c++) {
      for (d = 0 ; d < n; d++) {
         sum[c][d] = first[c][d] + second[c][d];
         printf("%d\t", sum[c][d]);
      }
      printf("\n");
   }

   return 0;
}
int main()
{ add();
}
```

Sample output :

```
 Enter the number of rows and columns of matrix
2 2
Enter the elements of first matrix
2 4
6 8
Enter the elements of second matrix
2 4
6 8
Sum of entered matrices:-
4       8
12      16
```

Result :

The program is executed and is the result is found to be right.

Ex no. 1 c	MATRIX MULTIPLICATION
Date:	

AIM:

To find the resultant of multiplication of two matix.

Algorithm:

Step 1 : Start

Step 2 : Declare required variables and get input from the user.

Step 3 : invoke the function multiply(). Which multiplies two matix.

Step 4 : The function multiply() should contain an nested for loop which calculates multiplication of matix using formula c[i][j]

Step 5 :

Source code:

```
#include <stdio.h>
void take_data(int a[][10], int b[][10], int r1,int c1, int r2, int c2);
void multiplication(int a[][10],int b[][10],int mult[][10],int r1,int c1,int r2,int c2);
void display(int mult[][10], int r1, int c2);
int main()
{
   int a[10][10], b[10][10], mult[10][10], r1, c1, r2, c2, i, j, k;
   printf("Enter rows and column for first matrix: ");
   scanf("%d%d", &r1, &c1);
   printf("Enter rows and column for second matrix: ");
   scanf("%d%d",&r2, &c2);
   while (c1!=r2)
   {
      printf("Error! column of first matrix not equal to row of second.\n");
      printf("Enter rows and column for first matrix: ");
```

```
     scanf("%d%d", &r1, &c1);
       printf("Enter rows and column for second matrix: ");
       scanf("%d%d",&r2, &c2);
   }
   take_data(a,b,r1,c1,r2,c2);  /* Function to take matices data */
   multiplication(a,b,mult,r1,c1,r2,c2); /* Function to multiply two matrices. */
   display(mult,r1,c2); /* Function to display resultant matrix after multiplication. */
   return 0;
}

void take_data(int a[][10], int b[][10], int r1,int c1, int r2, int c2)
{
   int i,j;
   printf("\nEnter elements of matrix 1:\n");
   for(i=0; i<r1; ++i)
   for(j=0; j<c1; ++j)
   {
      printf("Enter elements a%d%d: ",i,j);
      scanf("%d",&a[i][j]);
   }

   printf("\nEnter elements of matrix 2:\n");
   for(i=0; i<r2; ++i)
   for(j=0; j<c2; ++j)
   {
      printf("Enter elements b%d%d: ",i,j);
      scanf("%d",&b[i][j]);
   }
}

void multiplication(int a[][10],int b[][10],int mult[][10],int r1,int c1,int r2,int c2)
{
    int i,j,k;

    for(i=0; i<r1; ++i)
    for(j=0; j<c2; ++j)
    {
       mult[i][j]=0;
    }

    for(i=0; i<r1; ++i)
    for(j=0; j<c2; ++j)
    for(k=0; k<c1; ++k)
    {
        mult[i][j]+=a[i][k]*b[k][j];
    }
}

void display(int mult[][10], int r1, int c2)
{
    int i, j;
    printf("\nOutput Matrix:\n");
    for(i=0; i<r1; ++i)
    for(j=0; j<c2; ++j)
    {
        printf("%d  ",mult[i][j]);
        if(j==c2-1)
            printf("\n\n");
    }
}
```

Sample output :

```
 Enter rows and column for first matrix: 2 2
Enter rows and column for second matrix: 2 2

Enter elements of matrix 1:
Enter elements a00: 2
Enter elements a01: 4
Enter elements a10: 6
Enter elements a11: 8

Enter elements of matrix 2:
Enter elements b00: 2
Enter elements b01: 4
Enter elements b10: 6
Enter elements b11: 8

Output Matrix:
28  40


60    88
```

Result :

The program is executed and is the result is found to be right.

Ex no. 1 d	MATRIX TRANSPOSE
Date:	

AIM:

To find the transpose of the given matix.

Algorithm:

```
Step 1 : Start
Step 2 :  To find the resultant of multiplication of two matix.
Step 3 :  Transpose the given matix using the formula matrix[i] [j]=transpose[j] [i] in a for loop.
Step 4 :  Print the Transposed matrix.
Step 5 :  Stop.
```

Source code:

```
#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\n");
  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];
  for (c = 0; c < n; c++) {
    for (d = 0; d < m; d++)
      printf("%d\t",transpose[c][d]);
    printf("\n");
  }

  return 0;
}
```

Sample 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
```

Result :

The program is executed and is the result is found to be right.