

Output

The value of arr[0] is: 10

The value of arr[1] is: 38

The value of arr[2] is: 23

The value of arr[3] is: 56

The value of arr[4] is: 76

Program-1

//WAP to show compile time Initialization

```
#include <stdio.h>
#include <conio.h>
```

```
void main()
```

```
{  
    int arr[10] = {10, 38, 23, 56, 76};  
    printf("The value of arr[0] is : %d", arr[0]);  
    printf("The value of arr[1] is : %d", arr[1]);  
    printf("The value of arr[2] is : %d", arr[2]);  
    printf("The value of arr[3] is : %d", arr[3]);  
    printf("The value of arr[4] is : %d", arr[4]);
```

```
    getch();  
}
```

Output

Enter the size of an array 5

Enter the elements : 21

23
34
45
43

Elements given by User :

21
23
34
45
43

Program - 2

//WAP to show run time initialization

```
#include <stdio.h>
#include <conio.h>
```

```
void main()
```

```
{  
    int arr[7], n, i;  
    printf("nEnter the size of an array");  
    scanf("%d", &n);  
    printf("nEnter the elements :");  
    for(i=0; i<n; i++)  
    {  
        scanf("n%d", &arr[i]);  
    }  
    printf("nElements given by User :");  
    for(i=0; i<n; i++)  
    {  
        printf("n%d", arr[i]);  
    }  
}
```

Output

Input no. of elements : 5
Array elements are :

2
6
8
10
12

Array after reversing :

12
10
8
6
2

Program- 3

//WAP to reverse an array

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int arr[10], i, n;
    printf("Input no. of elements : ");
    scanf("%d", &n);
    printf("Array elements are : \n");
    for(i=0; i<n; i++)
    {
        scanf("%d", &arr[i]);
    }
    printf("Array after reversing : \n");
    for(i=n-1; i>=0; i--)
    {
        printf("\n%d", arr[i]);
    }
}
```

Enter no. of elements : 5

Enter elements: 45

98

65

34

21

Array elements enter by you:

45

98

65

34

21

Program - 4

//WAP to input and display the arrays by using function

```
#include <stdio.h>
#include <conio.h>
void input(int m[], int n);
void display(int m[], int n);

void input(int m[], int n)
{
    int i;
    for(i=0; i<n; i++)
        { scanf("%d", &m[i]); }
}

void display(int m[], int n)
{
    int i;
    for(i=0; i<n; i++)
        { printf("\n%d", m[i]); }
}

int main()
{
    int m[10], n;
    printf("Enter no. of elements: ");
    scanf("%d", &n);
    input(m, n);
    printf("Array elements enter by you: ");
    display(m, n);
    getch();
}
```

Date _____

Page No.: _____

Output

Enter no. of elements: 5

Enter elements:

21

98

54

45

3

Array elements enter by you:

21

98

54

45

5

Largest number is = 98

Smallest number is = 3

Date _____

Page No.: _____

Program - 5

//WAP to show max and min elements with the help of functions.

```
#include <stdio.h>
#include <conio.h>
void input(int arr[], int n)
{
    int i;
    printf("Enter elements:\n");
    for(i=0; i<n; i++)
    {
        scanf("\n%d", &arr[i]);
    }
}
```

```
void display(int arr[], int n)
{
    int i;
    printf("Array elements enter by you : ");
    for(i=0; i<n; i++)
    {
        printf("\n %d", arr[i]);
    }
}
```

```
void minmax(int arr[], int n)
{
    int i, max, min;
    max = arr[0];
    for(i=0; i<n; i++)
    {
        if (arr[i] > max)
        {
            max = arr[i];
        }
    }
}
```

"No real change in history has ever been achieved by discussions." —Subhash Chandra Bose

```
min = arr[0];
for(i=0; i<n; i++)
{
    if(arr[i] < min)
    {
        min = arr[i];
    }
}
```

```
printf("Largest number is = %.d", max);
printf("Smallest number is = %.d", min);
```

{

```
int main()
```

{

```
int arr[10], n, b[10], s[10];
printf("Enter no. of elements : ");
scanf("%d", &n);
input(arr, n);
display(arr, n);
minmax(arr, n);
```

3

Output

Enter the No. of Elements : 5

Input array elements : 3

4
5
6
8

Even Numbers are :

4
6
8

Odd Numbers are :

3
5

Program-6

//WAP to find odd and even element with the help of functions.

```
#include <stdio.h>
#include <conio.h>
```

```
void input(int arr[10], int n)
{
    int i;
    printf("Input array elements : ");
    for(i=0; i<n; i++)
    {
        scanf("%d", &arr[i]);
    }
}
```

```
void display(int arr[10], int n)
```

```

int i;
printf("Array Elements Enter By You : ");
for(i=0; i<n; i++)
{
    printf("\n%d", arr[i]);
}
```

```
void evenodd(int arr[10], int even[10], int
    odd[10], int n)
```

```

int i, j=0, k=0;
for(i=0; i<n; i++)
{
    if(arr[i] % 2 == 0)
        even[j] = arr[i];
    else
        odd[k] = arr[i];
    j++;
    k++;
}
```

```

if( arr[i] % 2 == 0 )
{
    even[j] = arr[i];
    j++;
}
else
{
    odd[k] = arr[i];
    k++;
}

```

```

printf("In Even Numbers are: ");
for(i=0; i<j; i++)
    printf("%d", even[i]);
printf("In Odd Numbers are ");
for(j=0; i<k; i++)
    printf("%d", odd[i]);
}

```

```

int main()
{

```

```

    int arr[10], n, even[10], odd[10];
    printf("Enter the No. of Elements: ");
    scanf("%d", &n);
    input(arr, n);
    display(arr, n);
    evenodd(arr, even, odd, n);
}

```

Output

Enter No. of elements : 5

Enter elements :

2
3
4
5
6

Array elements enter by you :

2
3
4
5
6

Elements of an Array after adding 10:

12
13
14
15
16

Program 7

//WAP to add 10 in all the elements of an array with the help of functions.

```
#include <stdio.h>
#include <conio.h>
```

```
void input(int arr[], int n)
{
```

```
    int i;
    printf("Enter elements :\n");
    for (i=0; i<n; i++)
    {
        scanf("%d", &arr[i]);
    }
```

```
void display(int arr[], int n)
{
```

```
    int i;
    printf("Array elements enter by you :\n");
    for (i=0; i<n; i++)
    {
        printf("\n%d", arr[i]);
    }
```

```
void add(int arr[], int n)
{
```

```
    int i;
    printf("\n Elements of an Array after adding
10:\n");
```

```
{ for(i=0; i<n; i++)
    {
        arr[i] = arr[i]+10;
        printf("In %d", arr[i]);
    }
}
```

```
int main()
{
    int arr[10], n;
    printf("Enter no. of elements : ");
    scanf("%d", &n);
    input(arr, n);
    display(arr, n);
    add(arr, n);
}
```

```
1 //wap to show linear search by using function
2 #include <stdio.h>
3 void input(int arr[], int n);
4 void display(int arr[], int n);
5 void linearSearch(int arr[], int n, int target);I
6
7 int main() {
8     int n, k, arr[10], target;
9     printf("Enter the number of elements in the array: ");
10    scanf("%d",&n);
11    printf("Enter the elements of the array:\n");
12    input(arr, n);
13    printf("The array is: ");
14    display(arr, n);
15    printf("Enter the element to search for: ");
16    scanf("%d",&target);
17    linearSearch(arr, n, target);
18
19    return 0;
20 }
21
22 void input(int arr[], int n) {
23     for (int i = 0; i<n; i++) {
24         printf("Element [%d]: ", i);
25         scanf("%d", &arr[i]);
26     }
27 }
```

```
29 void display(int arr[], int n) {  
30     for (int i = 0; i <n; i++) {  
31         printf("%d ", arr[i]);  
32     }  
33     printf("\n");  
34 }  
35  
36 void linearSearch(int arr[], int n, int target)  
37 {  
38     int i;  
39     for (int i=0; i <n; i++)  
40         if (arr[i]==target)  
41             printf("Element %d found at index %d.\n", target, i );  
42  
43 }  
44
```

OUTPUT

Enter the number of elements in the array: 5

Enter the elements of the array:

Element [0]: 77

Element [1]: 99

Element [2]: 33

Element [3]: 55

Element [4]: 11

The array is: 77 99 33 55 11

Enter the element to search for: 99

Element 99 found at index 1.

PS D:\C++ Program> █

```
1 //wap to show binary search by function
2 #include <stdio.h>
3 void inputArray(int arr[], int size);
4 void displayArray(int arr[], int size);
5 int binarySearch(int arr[], int size, int target);
6
7 int main() {
8     int size, target, index;
9     printf("Enter the number of elements in the array: ");
10    scanf("%d", &size);
11    int arr[size];
12    printf("Enter the elements of the sorted array (in ascending order):\n");
13    inputArray(arr, size);
14    printf("The array is: ");
15    displayArray(arr, size);
16    printf("Enter the element to search for: ");
17    scanf("%d", &target);
18    index = binarySearch(arr, size, target);
19    if (index != -1)
20    { printf("Element %d found at index %d.\n", target, index); }
21    else
22    { printf("Element %d not found in the array.\n", target); }
23    return 0; }
```

```
25 void inputArray(int arr[], int size) {
26     for (int i = 0; i < size; i++) {
27         printf("Element [%d]: ", i);
28         scanf("%d", &arr[i]);
29     }
30 }
31
32 void displayArray(int arr[], int size) {
33     for (int i = 0; i < size; i++) {
34         printf("%d ", arr[i]);
35     }
36     printf("\n");
37 }
38
39 int binarySearch(int arr[], int size, int target) {
40     int low = 0;
41     int high = size - 1;
42     int mid;
43
44     while (low <= high) {
45         mid = (low + high) / 2;
```

```
38
39  int binarySearch(int arr[], int size, int target) {
40      int low = 0;
41      int high = size - 1;
42      int mid;
43
44      while (low <= high) {
45          mid = (low + high) / 2;
46
47          if (arr[mid] == target) {
48              return mid;
49          } else {
50              if (arr[mid] < target) {
51                  low = mid + 1;
52              } else {
53                  high = mid - 1;
54              }
55          }
56      }
57      return -1; // Target not found
58 }
```

OUTPUT

```
Enter the number of elements in the array: 5
Enter the elements of the array (in ascending order):
Element [0]: 3
Element [1]: 5
Element [2]: 7
Element [3]: 9
Element [4]: 12
The array is: 3 5 7 9 12
Enter the element to search for: 7
Element 7 found at index 2.
```

#wap to show bubblesort by using function

BUBBLE.C

```
#include<stdio.h>
#include<conio.h>

void input(int arr[],int n);
void display(int arr[],int n);
void bubblesort(int arr[],int n);
void main()
{
    int arr[10],n;
    printf("enter the size of array");
    scanf("%d",&n);
    input(arr,n);
    printf("\n the array is \n");
    display(arr,n);
    printf("\n after the sorting array is \n");
    bubblesort(arr,n);
    display(arr,n);
    getch();
}

void input(int arr[],int n)
{
```

6:48

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 M

HCL

File Edit Search Run Compile Debug Project Options Window

[] BUBBLE.C

```
void input(int arr[],int n)
{
    int i;
    for(i=0;i<n;i++)
        printf("enter the array%d element",i);
    scanf("%d",&arr[i]);
}

void display(int arr[],int n)
{
    int i;
    for(i=0;i<n;i++)
        printf("\n%d",arr[i]);
}

void bubblesort(int arr[],int n)
{
    int i,j,temp;
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-1-i;j++)
        {
            if(arr[j]>arr[j+1])
            {
                temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
}
```

Activate W
Go to Settings

40:48

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10

BUBBLE.C

```
[*]=====
{int i;
for(i=0; i<n; i++)
printf("\n%d", arr[i]);
}

void bubblesort(int arr[], int n)
{
int i, j, temp;
for(i=0; i<n-1; i++)
{
for(j=0; j<n-1-i; j++)
{
if(arr[j]>arr[j+1])
{
temp=arr[j];
arr[j]=arr[j+1];
arr[j+1]=temp;
}
}
}
}
=====
```

48:48

OUTPUT

Enter the number of elements in the array: 5

Enter the elements of the array:

Element [0]: 3

Element [1]: 88

Element [2]: 3

Element [3]: 91

Element [4]: 66

The array is: 3 88 3 91 66

Enter the element to search for: 66

Element 66 found at index 4.

PS D:\C++ Program> █

```
- File Edit Search Run Compile Debug Project  
[ ] GSELECTI.C  
//Program for selection sort  
#include<stdio.h>  
#include<conio.h>  
void input(int arr[], int n);  
void selectionsort(int arr[],int n);  
void display(int arr[],int n);  
void main()  
{  
    int arr[10],n;  
    clrscr();  
    printf("Enter the size of array");  
    scanf("%d",&n);  
    input(arr,n);  
    printf("\n The array is \n");  
    display(arr,n);  
    printf("\n After Sorting the array is \n");  
    selectionsort(arr,n);  
    display(arr,n);  
    getch();  
}
```

```
getch();
}
void selectionsort(int arr[], int n)
{
    int i, j, temp, min_idx;
    for(i=0; i<n; i++)
    {
        min_idx=i;
        for(j=i+1; j<n; j++)
        {
            if(arr[j]<arr[min_idx])
                {min_idx=j;}
        }
        temp=arr[i];
        arr[i]=arr[min_idx];
        arr[min_idx]=temp;
    }
}
```

```
void input(int arr[], int n)
{
int i;
for(i=0; i<n; i++)
{
    printf("Enter the array %d ", i);
    scanf("%d", &arr[i]);
}
}

void display(int arr[], int n)
{
int i;
for(i=0; i<n; i++)
{
    printf("\n%d", arr[i]);
}
}
```

* 55:9 =□

OUTPUT

Enter the size of array: 5

Enter the array

99

66

55

21

10

The array is

99 66 55 21 10

After sorting array is

10 21 55 66 99

```
1 //wap to show insertion sort
2
3 #include <stdio.h>
4 void insertionSort(int arr[], int n);
5 void printArray(int arr[], int n);
6
7 int main()
8 {
9     int arr[] = {12, 11, 13, 5, 6};
10    int n = sizeof(arr) / sizeof(arr[0]);
11    printf("Original array: \n");
12    printArray(arr, n);
13    insertionSort(arr, n);
14    printf("Sorted array: \n");
15    printArray(arr, n);
16    return 0;
17 }
18
```

```
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i-1;
        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }

}
void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
```

Output

Original array:

12 11 13 5 6

Sorted array:

5 6 11 12 13

```
1 //wap to for addition of two matrices
2 #include <stdio.h>
3 void inputMatrix(int matrix[10][10], int row, int col);
4 void displayMatrix(int matrix[10][10], int row, int col);
5 void addMatrices(int matrix1[10][10], int matrix2[10][10], int result[10][10], int row, int col);
6 ....
7 int main()
8 {
9     int row, col;
10    int matrix1[10][10], matrix2[10][10], result[10][10];
11    printf("Enter the number of rows and columns: ");
12    scanf("%d %d", &row, &col);
13    printf("Input for first matrix:\n");
14    inputMatrix(matrix1, row, col);
15    printf("Input for second matrix:\n");
16    inputMatrix(matrix2, row, col);
17    addMatrices(matrix1, matrix2, result, row, col);
18    printf("First matrix:\n");
19    displayMatrix(matrix1, row, col);
20    printf("Second matrix:\n");
21    displayMatrix(matrix2, row, col);
22    printf("Resultant matrix after addition:\n");
23    displayMatrix(result, row, col);
24    return 0;
25 }
```



x ⑤ △ 0 C++ Program ✘ Debug ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ 0 ⌂ ⑤ Ln 50, Col 1 Spaces: 4 UTF-8 CRLF {} C windows-gcc-x86

```
26
27 void inputMatrix(int matrix[10][10], int row, int col)
28 {
29     printf("Enter elements of the matrix:\n");
30     for (int i = 0; i < row; i++) {
31         for (int j = 0; j < col; j++) {
32             printf("Element [%d][%d]: ", i, j);
33             scanf("%d", &matrix[i][j]);
34         } } }
35 void displayMatrix(int matrix[10][10], int row, int col)
36 {
37     for (int i = 0; i < row; i++) {
38         for (int j = 0; j < col; j++) {
39             printf("%d ", matrix[i][j]);
40         }
41         printf("\n"); } }
42 void addMatrices(int matrix1[10][10], int matrix2[][10], int result[10][10], int row, int col)
43 {
44     for (int i = 0; i < row; i++) {
45         for (int j = 0; j < col; j++) {
46             result[i][j] = matrix1[i][j] + matrix2[i][j];
47         }
48     }
49 }
50 }
```

13-B. (C++ Program) cd "D:\C++ Programs" , ln -s ./ addition.c & addition , ln -s ./

OUTPUT

Enter the number of rows and columns: 2

2

Input for first matrix:

Enter elements of the matrix:

Element [0][0]: 2

I

Element [0][1]: 4

Element [1][0]: 6

Element [1][1]: 8

Input for second matrix:

Enter elements of the matrix:

Element [0][0]: 1

Element [0][1]: 3

Element [1][0]: 5

Element [1][1]: 7

First matrix:

2 4

6 8

Second matrix:

1 3

5 7

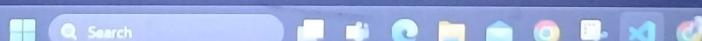
Resultant matrix after addition:

3 7

11 15

```
1 //wap to find transpose of a matrix
2 #include <stdio.h>
3 void inputMatrix(int matrix[][10], int row, int col);
4 void displayMatrix(int matrix[][10], int row, int col);
5 void transposeMatrix(int matrix[][10], int transposed[][10], int row, int col);
6
7 int main() {
8     int row, col;
9     int matrix[10][10], transposed[10][10];
10    printf("Enter the number of rows and columns: ");
11    scanf("%d %d", &row, &col);
12    printf("Input for matrix:\n");
13    inputMatrix(matrix, row, col);
14    transposeMatrix(matrix, transposed, row, col);
15    printf("Original matrix:\n");
16    displayMatrix(matrix, row, col);
17    printf("Transposed matrix:\n");
18    displayMatrix(transposed, col, row);
19
20 }
```

29°C
Cloudy



ENG
IN 01-08-2024

```
void inputMatrix(int matrix[][10], int row, int col) {
    printf("Enter elements of the matrix:\n");
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &matrix[i][j]);
        }
    }
}

void displayMatrix(int matrix[][10], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++)
        { printf("%d ", matrix[i][j]); }
        printf("\n");
    }
}

void transposeMatrix(int matrix[][10], int transposed[][10], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            transposed[j][i] = matrix[i][j];
        }
    }
}
```

OUTPUT

Enter the number of rows and columns: 2

2

Input for matrix:

Enter elements of the matrix:

Element [0][0]: 9

Element [0][1]: 8

Element [1][0]: 7

Element [1][1]: 6

Original matrix:

9 8

7 6

Transposed matrix:

9 7

8 6

```
1 //wap to multiply two matrices
2 #include <stdio.h>
3 void input(int matrix[][10], int row, int col);
4 void display(int matrix[][10], int row, int col);
5 void multiply(int matrix1[][10], int matrix2[][10], int result[][10], int row1, int col1, int row2, int col2);
6
7 int main() {
8     int row1, col1, row2, col2;
9     int matrix1[10][10], matrix2[10][10], result[10][10];
10    printf("Enter the number of rows and columns for first matrix: ");
11    scanf("%d %d", &row1, &col1);
12    printf("Enter the number of rows and columns for second matrix: ");
13    scanf("%d %d", &row2, &col2);
14    printf("Input for first matrix:\n");
15    input(matrix1, row1, col1);
16    printf("Input for second matrix:\n");
17    input(matrix2, row2, col2);
18    multiply(matrix1, matrix2, result, row1, col1, row2, col2);
19    printf("First matrix:\n");
20    display(matrix1, row1, col1);
21    printf("Second matrix:\n");
22    display(matrix2, row2, col2);
23    printf("Resultant matrix after multiplication:\n");
24    display(result, row1, col2);
25    return 0;
26 }
27
```

```
27
28 void input(int matrix[][10], int row, int col) {
29     printf("Enter elements of the matrix:\n");
30     for (int i = 0; i < row; i++) {
31         for (int j = 0; j < col; j++) {
32             printf("Element [%d][%d]: ", i, j);
33             scanf("%d", &matrix[i][j]);
34         }
35     }
36 }
37
38 void display(int matrix[][10], int row, int col) {
39     for (int i = 0; i < row; i++) {
40         for (int j = 0; j < col; j++) {
41             printf("%d ", matrix[i][j]);
42         }
43         printf("\n");
44     }
45 }
46
47 void multiply(int matrix1[][10], int matrix2[][10], int result[][10], int row1, int col1, int row2, int col2) {
48     for (int i = 0; i < row1; i++) {
49         for (int j = 0; j < col2; j++) {
50             result[i][j] = 0;
51             for (int k = 0; k < col1; k++) {
52                 result[i][j] += matrix1[i][k] * matrix2[k][j];
53             }
54         }
55     }
56 }
```



OUTPUT

```
Enter the number of rows and columns for first matrix: 2
2
Enter the number of rows and columns for second matrix: 2
2
Input for first matrix:
Enter elements of the matrix:
Element [0][0]: 1
Element [0][1]: 3
Element [1][0]: 4
Element [1][1]: 6
Input for second matrix:
Enter elements of the matrix:
Element [0][0]: 9
Element [0][1]: 6
Element [1][0]: 5
Element [1][1]: 4
First matrix:
1 3
4 6
Second matrix:
9 6
5 4
Resultant matrix after multiplication:
24 18
66 48
```