Batch: C1-1	**Roll No.: 16010123012**
Experiment / assignment / tutorial No. 4	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Write a program in C to demonstrate use of arrays

AIM: Program to sort the 1D array in the ascending or descending order and then accept the element from user and insert in the same array at its correct place by keeping array sorted
Write a program to find the Transpose of a Matrix.

Expected OUTCOME of Experiment:
Apply basic concepts of C programming for problem solving.(CO1 and CO2).

Books/ Journals/ Websites referred:

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving , G. Michael Schneider ,Wiley India edition.
4. **http://cse.iitkgp.ac.in/~rkumar/pds-vlab/**

Problem Definition:

1. The program takes a 1D array and sorts it in the specified manner. The user enters an element and the same has to be inserted at the correct place in the sorted array.

2. Write a program to find the Transpose of a Matrix.

- Entered matrix:

1 4 0

-5 2 7

- Transpose of the matrix:

1 -5

4 2

0 7

Algorithm:

1.

1. Input size of array.
2. Declare an unsorted array.
3. Input elements into the array.
4. Print the array
5. Perform bubble sort on the array:
6. Print the sorted array
7. Input a new element, to be added into the array.
8. Add the new element to the end of the array.
9. Increase the size of the array
10. Print the array with the new element
11. Perform bubble sort on the array again
12. Print the final sorted array.

2.

1. Read the number of rows (r) and columns (c) for the matrix from the user.

2. Declare a 2D array 'matrix' of size r x c.

3. Read the elements of the matrix from the user.

4. Declare a 2D array 'transpose' of size c x r.

5. Iterate through each element of the original matrix:

   a. Assign the element to the corresponding position in the transpose matrix.

6. Display the original matrix.

7. Display the transpose matrix.

Implementation details:

1.

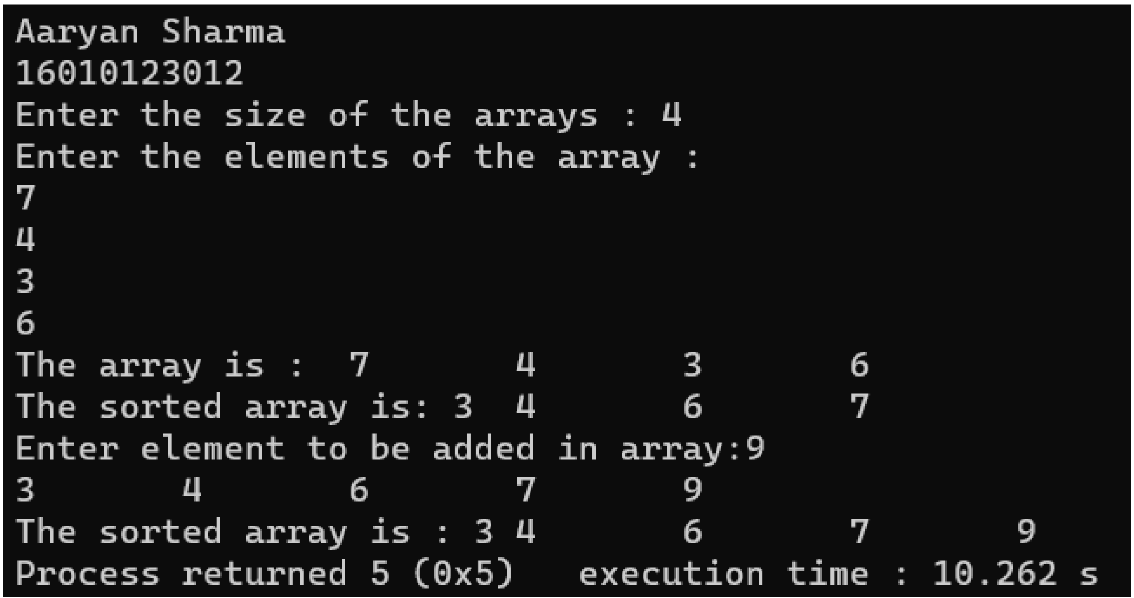
```
#include  <stdio.h>
void main()
{
printf("Aaryan Sharma\n");
printf("16010123012\n");

  int s, i, j, n, new;
  printf("Enter the size of the arrays : ");
  scanf("%d", &s);

  int arr[s];
  printf("Enter the elements of the array :\n");
  for (i = 0; i < s; i++)
  {
     scanf("%d", &arr[i]);
  }
  printf("The array is : ");
  for (i = 0; i < s; i++)
  {
    printf("\t%d",arr[i]);
  }
     for (i = 0; i < s; ++i)
     {
       for (j = i + 1; j < s; ++j)
       {
         if (arr[i] > arr[j])
         {
           n = arr[i];
         arr[i] = arr[j];
         arr[j] = n;
       }
     }
   }
 printf("\nThe sorted array is: ");
 for (i = 0; i < s; ++i)
 {
  printf("%d\t",arr[i]);
 }
 printf("\nEnter element to be added in array:");
 scanf("%d",&new);
 arr[s]=new;
 s++;
  for (i = 0; i < s; ++i)
 {
  printf("%d\t",arr[i]);
 }
  for (i = 0; i < s; ++i)
   {
     for (j = i + 1; j < s; ++j)
     {
       if (arr[i] > arr[j])
       {
         n = arr[i];
         arr[i] = arr[j];
         arr[j] = n;
       }
     }
   }
 printf("\nThe sorted array is : ");
 for (i = 0; i < s; ++i)
 {
  printf("%d\t",arr[i]);
 }
}
```

2.

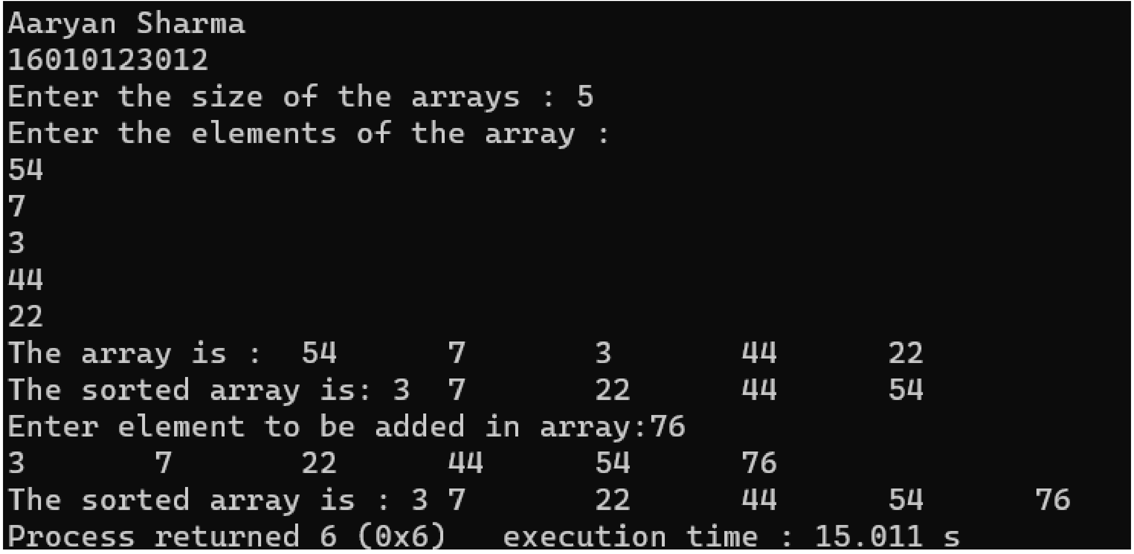
```
#include<stdio.h>
int main(void){
printf("Aaryan Sharma\n");
printf("16010123012\n");
int r,c,i,j;
printf("Enter number of rows and columns : ");
scanf("%d %d",&r,&c);
int a[r][c];
int t[r][c];

printf("Enter elements : ");
for(i=0;i<r;i++){
for(j=0;j<c;j++){
scanf("%d",&a[i][j]);
}
}
printf("Matrix is : \n");
for(i=0;i<r;i++){
for(j=0;j<c;j++){
printf("%d ", a[i][j]);
if (j == c - 1) {
printf("\n\n");
}
}
}
printf("\nTranspose matrix is : \n");
for(j=0;j<c;j++){
for(i=0;i<r;i++){
printf("%d ", a[i][j]);
if (i == r - 1) {
printf("\n\n");
}
}
}
}
```

Output(s):

1.

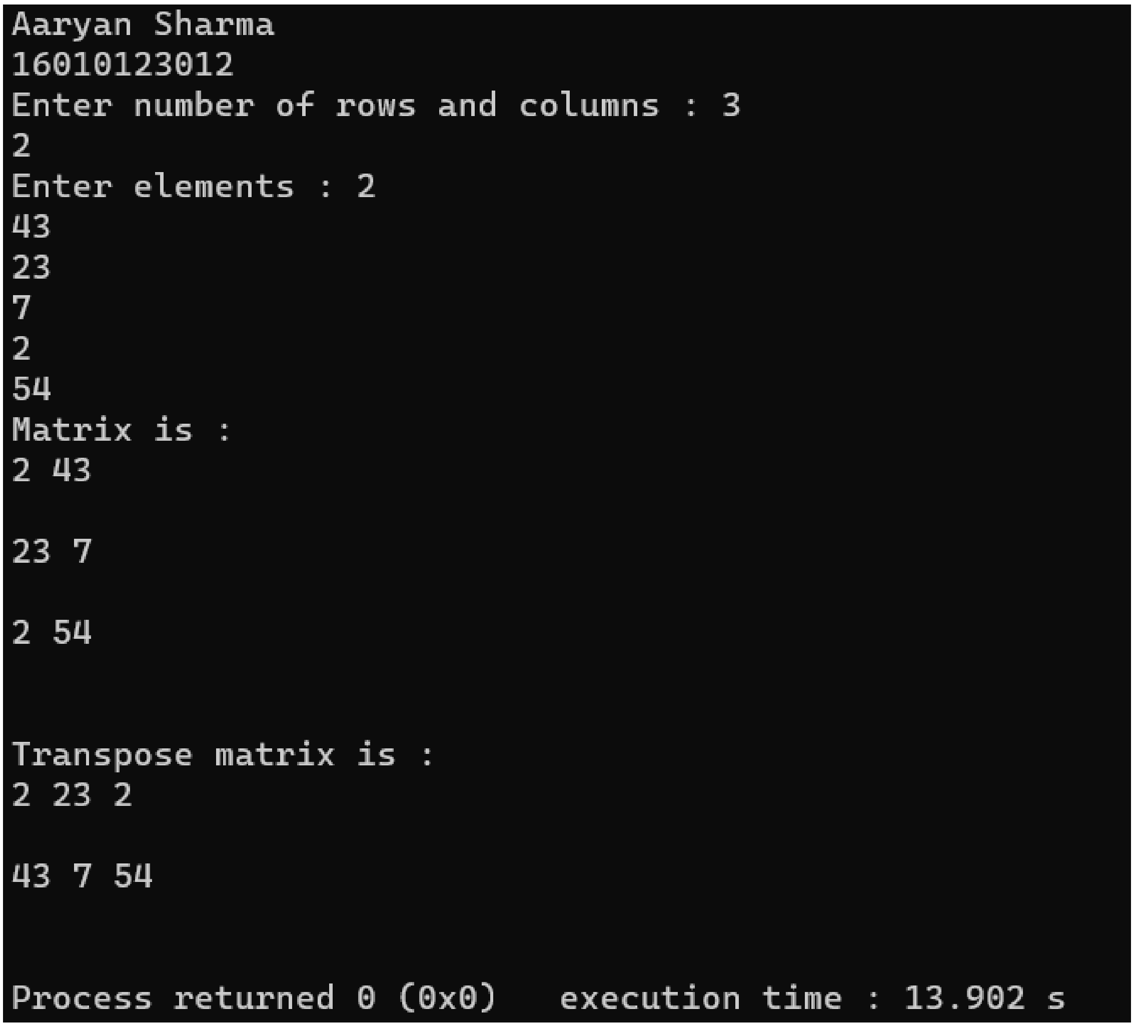


```
Aaryan Sharma
16010123012
Enter the size of the arrays : 4
Enter the elements of the array :
7
4
3
6
The array is :  7       4       3       6
The sorted array is: 3  4       6       7
Enter element to be added in array:9
3       4       6       7       9
The sorted array is : 3 4       6       7       9
Process returned 5 (0x5)   execution time : 10.262 s
```

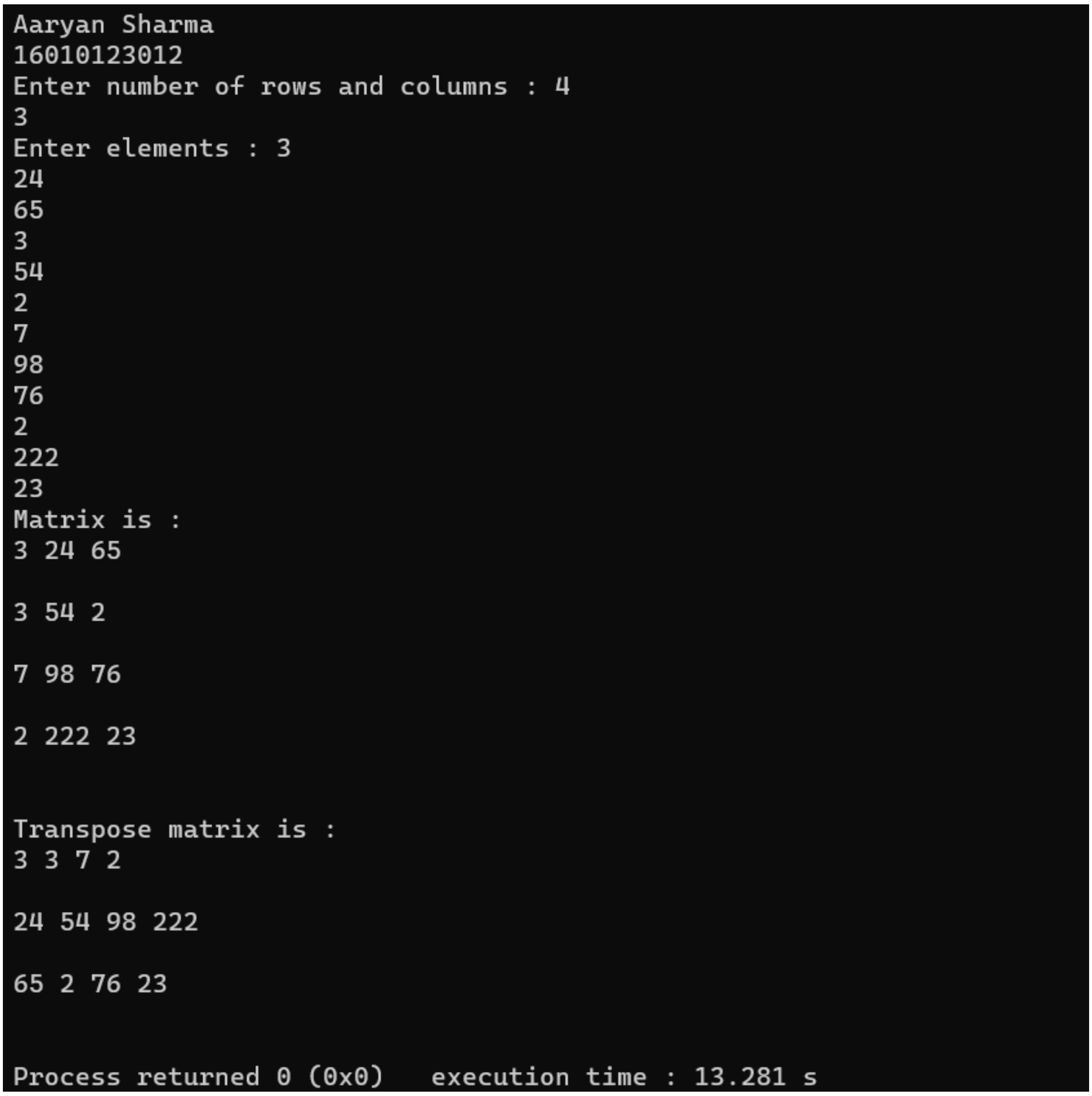


```
Aaryan Sharma
16010123012
Enter the size of the arrays : 5
Enter the elements of the array :
54
7
3
44
22
The array is :  54      7       3       44      22
The sorted array is: 3  7       22      44      54
Enter element to be added in array:76
3       7       22      44      54      76
The sorted array is : 3 7       22      44      54      76
Process returned 6 (0x6)   execution time : 15.011 s
```

2.



```
Aaryan Sharma
16010123012
Enter number of rows and columns : 3
2
Enter elements : 2
43
23
7
2
54
Matrix is :
2 43

23 7

2 54


Transpose matrix is :
2 23 2

43 7 54


Process returned 0 (0x0)   execution time : 13.902 s
```



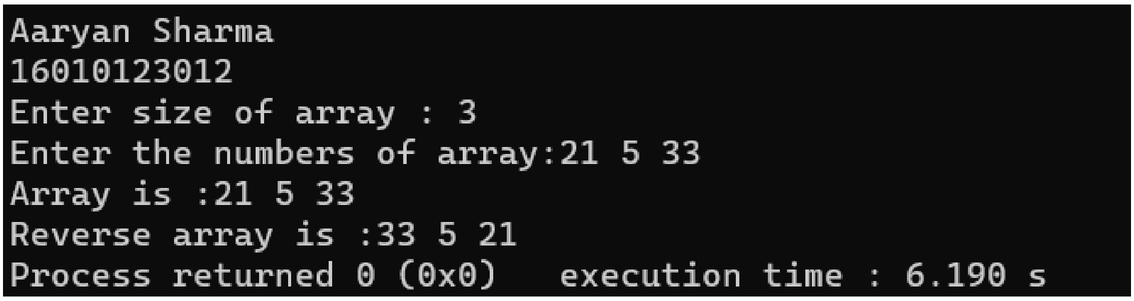
```
Aaryan Sharma
16010123012
Enter number of rows and columns : 4
3
Enter elements : 3
24
65
3
54
2
7
98
76
2
222
23
Matrix is :
3 24 65

3 54 2

7 98 76

2 222 23


Transpose matrix is :
3 3 7 2

24 54 98 222

65 2 76 23


Process returned 0 (0x0)   execution time : 13.281 s
```

Conclusion:
We have successfully completed the experiment and learned to use 1D and 2D arrays.

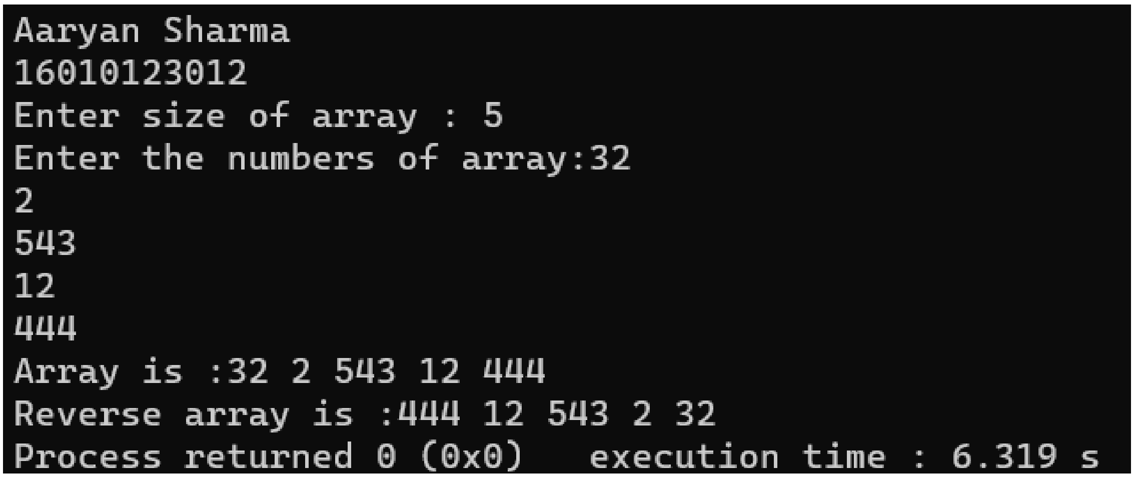
Post Lab Questions

1. Write a program to enter n numbers, store them in an array and rearrange the array in the reverse order.

```
#include<stdio.h>
int main(){
printf("Aaryan Sharma\n");
printf("16010123012\n");
int n,i,j;
printf("Enter size of array : ");
scanf("%d",&n);

int a[n];
int b[n];
printf("Enter the numbers of array:");
for(i=0;i<n;i++){
  scanf("%d",&a[i]);
}
printf("Array is :");
for(i=0;i<n;i++){
  printf("%d ",a[i]);
}
j=n-1;
for(i=0;i<n;i++){
  b[i]=a[j];
  j--;
}
printf("\nReverse array is :");
for(i=0;i<n;i++){
  printf("%d ",b[i]);
}
}
```



```
Aaryan Sharma
16010123012
Enter size of array : 3
Enter the numbers of array:21 5 33
Array is :21 5 33
Reverse array is :33 5 21
Process returned 0 (0x0)   execution time : 6.190 s
```



```
Aaryan Sharma
16010123012
Enter size of array : 5
Enter the numbers of array:32
2
543
12
444
Array is :32 2 543 12 444
Reverse array is :444 12 543 2 32
Process returned 0 (0x0)   execution time : 6.319 s
```

2. Write a program which performs the following tasks:

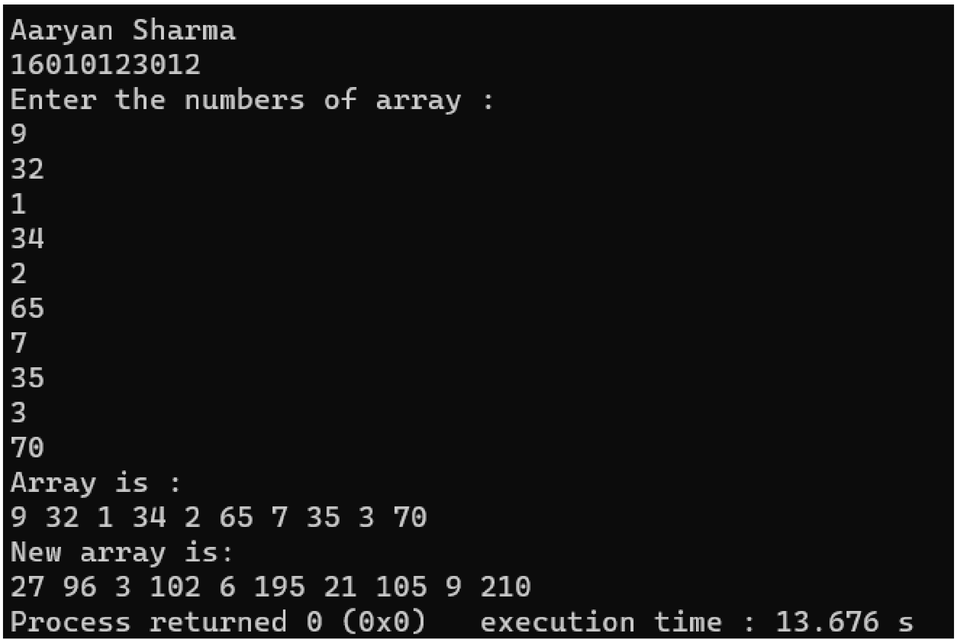
a) Initialize an integer array of 10 elements in main( )

b) Pass the entire array to a function modify( )

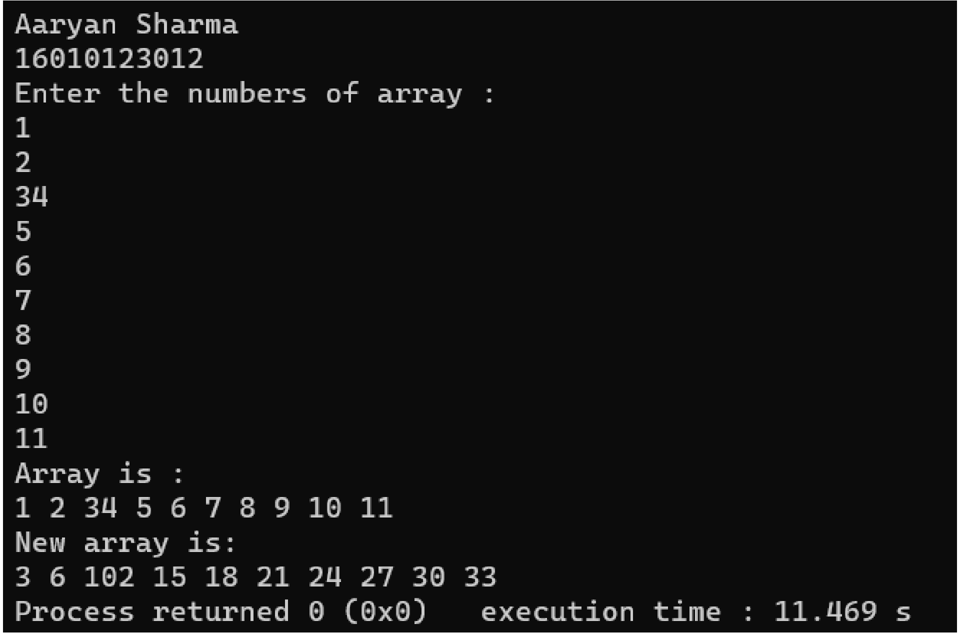
c) In modify( ) multiply each element of array by 3

d) Return the control to main( ) and print the new array elements in main( )

```
#include<stdio.h>
int main(){
printf("Aaryan Sharma\n");
printf("16010123012\n");

int i,arr[10];
printf("Enter the numbers of array :\n");
for(i=0;i<10;i++){
   scanf("%d",&arr[i]);
}
printf("Array is :\n");
for(i=0;i<10;i++){
   printf("%d ",arr[i]);
}
for(i=0;i<10;i++){
   arr[i]*=3;
}
printf("\nNew array is: \n");
for(i=0;i<10;i++){
   printf("%d ",arr[i]);
}
}
```



```
Aaryan Sharma
16010123012
Enter the numbers of array :
9
32
1
34
2
65
7
35
3
70
Array is :
9 32 1 34 2 65 7 35 3 70
New array is:
27 96 3 102 6 195 21 105 9 210
Process returned 0 (0x0)   execution time : 13.676 s
```



```
Aaryan Sharma
16010123012
Enter the numbers of array :
1
2
34
5
6
7
8
9
10
11
Array is :
1 2 34 5 6 7 8 9 10 11
New array is:
3 6 102 15 18 21 24 27 30 33
Process returned 0 (0x0)   execution time : 11.469 s
```

Date: 02/10/24 **Signature of faculty in-charge**