

1. Which of the statements is correct?
 - a) An array contains more than one element
 - b) All elements of array have to be of same data type
 - c) The size of array has to be declared upfront
 - d) All of the above

Solution: (d) All of the above

2. An integer array of size 15 is declared in a C program. The memory location of the first byte of the array is 2000. What will be the location of the 13th element of the array? Assume int data type takes 2 bytes of memory.
 - a) 2013
 - b) 2024
 - c) 2026
 - d) 2030

Solution: (b) Integer takes 2 bytes of memory. As the memory assignment to the elements is consecutive and the index starts from 0, the 13th element will be located at $2000 + (12 \times 2) = 2024$.

3. Which statement is correct?
 - a) An index or subscript in array is a positive integer
 - b) An index or subscript in array is a positive or negative integer
 - c) An index or subscript in array is a real number
 - d) None

Solution: (a) An index or subscript in an array is a positive integer. Negative or all real values are not allowed.

4. What will happen if in a C program you assign a value to an array element whose index exceeds the size of the array?
 - a) The element will be set to 0
 - b) The compiler will not give any error but the program may crash if some important data gets overwritten.
 - c) The compiler would report an error.
 - d) The array size would appropriately grow.

Solution: (b) The compiler will not give any error but the program may crash if some important data gets overwritten. The programmer has to be cautious in this case.

5. Which of the following statements is correct for the following code snippet?

```
int num[7];
num[7]=8;
```

- a) In the first statement 7 specifies a particular element, whereas in the second statement it specifies a type;
- b) In the first statement 7 specifies a particular element, whereas in the second statement it specifies the array size.

- c) In the first statement 7 specifies the array size, whereas in the second statement it specifies a particular element of array.
- d) In both the statement 7 specifies array size.

Solution: (c) The statement 'c' is correct, because `int num[7];` specifies the size of array and `num[7]=8;` designates the particular element (8th element) of the array.

6. What will be output of the following program?

```
int main()
{
    int i;
    int arr[3] = {3};
    for (i = 0; i < 3; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

- a) 3 followed by garbage values
- b) 3 0 0
- c) 3 1 1
- d) Syntax error

Solution: (b)

If an array is initialized with few elements, remaining elements will be initialized to 0. Therefore, 3 followed by 0, 0, will be printed.

7. What will be the output?

```
#include <stdio.h>
int main()
{
    int arr[]={1,2,3,4,5,6};
    int i,j,k;
    j=++arr[2];
    k=arr[1]++;
    i=arr[j++];
    printf("i=%d, j=%d, k=%d", i, j, k);
    return 0;
}
```

- a) i=5, j=5, k=2
- b) i=6, j=5, k=3
- c) i=6, j=4, k=2
- d) i=5, j=4, k=2

Solution: (a) `k=arr[1]++` due to post increment operation, assignment is done first. so it actually becomes `k=arr[1]=2`. `j=++arr[2]=++3=4`. `i=arr[j++]=arr[4]=5` (as its post increment hence assignment is done first). Due to post increment in `i=arr[j++]`, value of j is also incremented and finally becomes 5. So, finally `i=5, j=5, k=2`.

8. What will be the output after execution of the program?

```
#include <stdio.h>
int main()
{
    int i, a[4]={3,1,2,4}, result;
    result=a[0];
    for(i=1;i<4;i++)
    {
        if(result>a[i])
            continue;
        result=a[i];
    }
    printf("%d", result);
    return 0;
}
```

- a) 1
- b) 2
- c) 3
- d) 4

Solution: (d) The program finds the maximum element of an array. Hence, the output is 4.

9. What will be the output?

```
#include<stdio.h>
int main()
{
    int n = 2;
    int sum = 5;
    switch(n)
    {
        case 2: sum = sum-3;
        case 3: sum*=4;
        break;
        default:
            sum =0;
    }
    printf("%d", sum);
    return 0;
}
```

Solution: 8 (Short answer type)

$n=2$ therefore `switch(2)` i.e. case 2 will be executed. Inside case 2, sum becomes $\text{sum}-3 = 5-2 = 2$. As there is no break statement after case 2, therefore case 3 is also executed. Inside case 3, sum becomes $\text{sum}*4 = 2*4=8$. After that the execution finds a break statement and comes out of the switch. So, finally 8 is printed.

10. Find the output of the following C program

```
#include<stdio.h>
int main()
{
    int a;
    int arr[5] = {1, 2, 3, 4, 5};
    arr[1] = ++arr[1];
```

```

    a = arr[1]++;
    arr[1] = arr[a++];
    printf("%d, %d", a, arr[1]);
    return 0;
}

```

- a) 5,4
- b) 5,5
- c) 4,4
- d) 3,4

Solution: (c)

The execution steps are as follows:

1. `arr[1] = ++arr[1];` □ `arr[1]=++2=3` so, `arr={ 1, 3, 3, 4, 5 }`
2. `a = arr[1]++;` □ `a=arr[1]=3` (due to post increment). `arr` remains the same as step 1.
3. `arr[1] = arr[a++];` □ `arr[1]=arr[a]=arr[3]=4`. `arr={ 1, 4, 3, 4, 5 }`. `a` is incremented to `3+1=4` after the assignment is done.
4. Finally, `a=4` and `arr[1]=4` are printed