

1. Which of the following statement(s) is/are correct?

```
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.

2. What is the right way to initialize array in C?

- a) `int arr{}={1,2,5,6,9}`
- b) `int arr[5]={1,2,5,6,9}`
- c) `int arr{5}={1,2,5,6,9}`
- d) `int arr()={1,2,5,6,9}`

Solution: (b)

3. An integer array of size 10 is declared in a C program. The memory location of the first byte of the array is 1000. What will be the location of the 7th element of the array? (integer takes 2 bytes of memory)
- a) 1014
 - b) 1012
 - c) 1010
 - d) 1016

Solution: (b) Integer takes two bytes of memory. As the memory assignment to the elements are consecutive and the index starts from 0, the 7th element will be located at $1000 + (6 \times 2) = 1012$.

4. Which of the statement is correct?

- a) An array contains more than one element
- b) All elements of array has 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

5. 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;
}

```

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

6. Which assignment is not valid for integer arrays in C programming? (arr, arr1 and arr2 are integer arrays)
- arr[4]={ 1,2,5,3,4};
 - arr[]={ 1,2,3,4}
 - arr1=arr2
 - All are valid assignment

Solution: (c) direct assignment from one array to another array is invalid in C language.

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;
}

```

- i=5, j=5, k=2
- i=6, j=5, k=3
- i=6, j=4, k=2
- 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 when the following code is executed.

```

#include <stdio.h>
int main()
{
    int a[6]={ 1,2,3,4,5,6};
    switch(sizeof(a))
    {
        case 1:
        case 2:
        case 3:
        case 4:
        case 5:
            printf("IIT KGP");
    }
}

```

```

        break;
    }
    printf("IIT MADRAS");
    return 0;
}

```

Solution: IIT MADRAS

Size of the array a is 6, which does not belong to any case number. So, only IIT MADRAS will be printed outside of the switch.

9. How many 'a' will be printed when the following code is executed?

```

#include <stdio.h>
int main()
{
    int i = 0;
    char c = 'a';
    while (i < 5)
    {
        i++;
        switch (c)
        {
            case 'a':
                printf("%c ", c);
                break;
        }
    }
    printf("a\n");
    return 0;
}

```

Solution: 6

Initially, i=0, which satisfies the while condition. Case 'a' is always executed inside the while loop for i=1 to i=5 i.e., 5 times. Finally, another 'a' will be printed that is outside of the while loop. Therefore, total 6 times 'a' 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

WEEK 6 ASSIGNMENT SOLUTION

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 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