

Week 6 Assignment & Solution

1. The index of an array can take
 - a) only integer
 - b) only float
 - c) both integer and float
 - d) character

Solution: (a) Only integer variables are allowed as array index.

2. Which of the following is correct statement to access 5th element in a array arr[] of size 50?
 - a) arr[5]
 - b) arr[4]
 - c) arr{5}
 - d) arr{4}

Solution: (b) arr[4] is the correct syntax to access the 5th element, as in C index starts from 0.

3. 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)

4. What will be the output of the C program?

```
#include<stdio.h>
int main()
{
    int arr[1] = {2};
    printf("%d", arr[0]);
    return 0;
}
```

Solution: 2 as arr[0] prints the first element of the array.

5. What will be the output of the C program?

```
#include<stdio.h>
int main()
{
    int i;
    int arr[5]={10,20,30,40,50};
    printf("%d ", arr[2]);
    printf("%d ", 2[arr]);
    printf("%d ", arr[-2]);
    return 0;
}
```

- a) 30 30 garbage-value
- b) 30 30 30
- c) 30 20 garbage-value
- d) 30 60 40

Week 6 Assignment & Solution

Solution: (a) `arr[2]` is the 3rd element of the array i.e. 30. Similarly `2[arr]` also prints the 3rd element of the array i.e. 30. `arr[-2]` prints the value at the address location at base address of `arr[]` minus $2 \times (\text{size of int})$. As we have not stored any value at that location it will take a garbage value.

6. An integer array of dimension 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? [Assuming the integer takes 2 bytes of memory]

- a) 2013
- b) 2024
- c) 2026
- d) 2030

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

7. What will be output?

```
#include <stdio.h>
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)

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

Week 6 Assignment & Solution

c) 3

d) 4

Solution: (a) The program finds the minimum element of an array. Hence, the output is 1.

9. What will the output?

```
#include <stdio.h>
int main()
{
    int arr[]={1,2,3,4,5,6};
    printf("%d", arr[10]);
    return 0;
}
```

a) Garbage value

b) 0

c) 1

d) Compiler dependent

Solution: (a) Garbage value.

Since array size is 6 but you are accessing 11th element which is not set, so it will print garbage value.

10. What will be the output?

```
#include <stdio.h>
int main()
{
    int arr1[]={1,2,3,4,5,6};
    int arr2[]={5,2,2,7,1,0};
    int arr3[];
    arr3[]=arr1[]+arr2[];
    printf("%d", arr3[]);
    return 0;
}
```

a) 6,4,5,11,6,6

b) 1,2,3,4,5,6

c) 5,2,2,7,1,0

d) Error

Solution: (d) Error.

11. Which assignment is not valid for integer arrays in C programming? (arr, arr1 and arr2 are integer arrays)

a) arr[4]={1,2.5,3,4};

b) arr[]={1,2,3,4};

c) arr1=arr2;

d) All are valid assignment

Week 6 Assignment & Solution

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

12. To compare two arrays, we can use
- a) Comparison operator '==' directly on arrays
 - b) Use switch case
 - c) Using for loop
 - d) Using ternary operator on arrays

Solution: (c) We can use for loop and equality check operator on each element of the arrays to compare.

13. 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)

14. 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

Week 6 Assignment & Solution

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 $k = arr[1] = 2$. $j = ++arr[2] = ++3 = 4$. $i = arr[j++] = arr[4++] = arr[4] = 5$ (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 $i = 5$, $j = 5$, $k = 2$.

15. What will be the output of the C program?

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

- a) 1
- b) 2
- c) Compilation error

Solution (c) array can't be declared in #define pre-processor.