

Week 6 Assignment Solution

1. What is an array in C?
 - a) A collection of similar data elements with the same data type.
 - b) A built-in function that performs mathematical calculations.
 - c) A keyword used for declaring variables.
 - d) A data type used to store characters only.

Solution: a) A collection of similar data elements with the same data type.

Explanation: An array in C is a collection of elements of the same data type stored in contiguous memory locations. Each element in the array can be accessed using an index, which starts from 0 and goes up to (array_size - 1).

2. What is the index of the first element in an array?
 - a) 0
 - b) 1
 - c) -1
 - d) The index can vary depending on the array size.

Solution: a) 0

Explanation: In C, array indices start from 0. Therefore, the index of the first element in an array is 0, the second element's index is 1, and so on.

3. Which loop is commonly used to iterate through all elements of an array in C?
 - a) for loop
 - b) while loop
 - c) do-while loop
 - d) switch loop

Solution: a) for loop

Explanation: The "for" loop is commonly used to iterate through all elements of an array in C. It allows precise control over the loop variable and is well-suited for iterating over a range of elements, as required in array traversal.

4. 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? Assume int 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 is consecutive and the index starts from 0, the 13th element will be located at $2000 + (12 \times 2) = 2024$.

5. How can you find the sum of all elements in a 1D array "arr" with 5 elements using loop in C?
 - a) `sum = arr[0] + arr[1] + arr[2] + arr[3] + arr[4];`

Week 6 Assignment Solution

- b) `sum = arr[5];`
- c) `for (int i = 0; i <= 5; i++) { sum += arr[i]; }`
- d) `for (int i = 0; i < 5; i++) { sum += arr[i]; }`

Solution: d) `for (int i = 0; i < 5; i++) { sum += arr[i]; }`

Explanation: To find the sum of all elements in a 1D array "arr" with 5 elements, you can use a "for" loop to iterate through the array and add each element to the "sum" variable. The code provided correctly calculates the sum of all elements.

6. What is the output of the following code?

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

- a) 1 3 5
- b) 1 2 3 4 5
- c) 1 2 3
- d) 1 4

Solution: a) 1 3 5

Explanation: The provided code uses a "while" loop to print elements of the array "arr" with an increment of 2 for the loop counter "i." The loop starts at index 0 and prints elements at indices 0, 2, and 4. The output will be "1 3 5."

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

Week 6 Assignment Solution

- 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;
}
```

Week 6 Assignment Solution

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