Week 6: Assignment Solution 2023

1. 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 of the above

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

2. Which of the following is the correct way to declare a one-dimensional integer array named "myArray" with 5 elements in C?

    a) int myArray(5);

    b) int myArray[5];

    c) int myArray = [5];

    d) int myArray = {5};

Solution (b) int myArray[5]

3. The elements of array are stored in contiguous memory due to
    a) This way computers can keep track of only the address of the first element and the addresses of other elements can be calculated.
    b) The architecture of computer does not allow arrays to store other than serially
    c) Both (a) and (b) are true
    d) None of these are true

Solution: (a) This way computers can keep track only the address of the first element and the addresses of other elements can be calculated easily.

4. What is the output of the following code snippet in C?

```
int myArray[] = {1, 2, 3, 4, 5};
printf("%d", myArray[5]);
```

    a) 0
    b) 1
    c) 5
    d) The code will result in a runtime error.

Solution: (d) The code will result in a runtime error. This is because the array has 5 elements, but the highest index that can be accessed is 4 (since the indices start at 0). Accessing an index outside the bounds of the array can result in undefined behaviour or a segmentation fault.

5. What is the output of the code snippet?

```
int myArray[5] = {10, 20, 30, 40, 50};
int sum = 0;
for (int i = 0; i < 5; i++) {
sum += myArray[i];
}
printf("The sum of the array is %d", sum);
```

Solution: 150
The sum of the array is 150. This code declares an integer array with 5 elements and initializes its values to 10, 20, 30, 40, and 50. The variable sum is initialized to 0, and a for loop iterates over the array, adding each element to the sum variable. After the loop completes, the printf statement prints out the value of sum. Since the values of the array are 10, 20, 30, 40, and 50, the sum of the array is 150. Therefore, the output is "The sum of the array is 150".

6. What is the maximum dimension of an array allowed in C?
   a) 10
   b) 100
   c) 1000
   d) No such limit exists

Solution: (d) No such theoretical limit exists. It essentially limited by the memory and the compiler.

7. What is the output of the following C code?

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

   a) 1 followed by two garbage values
   b) 1 1 1
   c) 1 0 0
   d) 0 0 0

Solution: (c) Any uninitialized value in array is considered to be zero by default in C programming. Thus the content of arr is 1 0 0.

8. A one-dimensional array X has indices 1....50. Each element is an int. The array is stored at location 1000 decimal. The starting address of X[27] is _____(assume int takes 4 bytes).

Solution: Start address of the element = base address of array + number of elements before it * size of each element = 1000 + 26 * 4 = 1104

9. Which of the following statements finds the minimum value in an integer array named "myArray" in C?

   a) int minVal = myArray[0];

   b)
```
int minVal = 0;
for (int i = 0; i < 5; i++)
{
  if (myArray[i] > minVal)
  {
        minVal = myArray[i];

  } }
```

   c)
```
int minVal = INT_MAX;
for (int i = 0; i < 5; i++)
{
    if (myArray[i] < minVal)
    {
      minVal = myArray[i];
    }

}
```

   d) int minVal = myArray[4];

Answer: c) int minVal = INT_MAX; for (int i = 0; i < 5; i++) {if (myArray[i] < minVal) { minVal = myArray[i]; } }.

This code initializes a variable called minVal to the maximum possible value of an integer (defined by the INT_MAX constant), and then loops through the array to find the minimum value. If a value in the array is smaller than the current minimum value, it becomes the new minimum value. At the end of the loop, minVal contains the smallest value in the array.

10. 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++]=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.