ASSIGNMENT - 6 SOLUTION

1. 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.

2. 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.

3. Which of the following statement is correct for 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.

4. 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

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

5. 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 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 13^{th} element will be located at 2000+(12×2) =2024.

6. To compare two arrays, we can use
   a) Comparison operator '==' directly on arrays
   b) Using "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.

7. 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 is 1. The program finds the minimum element of an array. Hence, the output is 1.

8. What is the output of the following C program?

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

   a) 3
   b) 4
   c) No output
   d) Compilation error

Solution: (d) The size of the array does not match with the dimension of the array. Thus the compiler will show error.

9. 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)

10. Predict the output of the following code.

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

a) 0
b) 1
c) 5
d) Syntax error

Solution: (c) printf("%d", 0[arr]); It prints the first element value of the variable arr. Hence, the output is 5.

11. 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.

12. Array elements are stored in memory in the following order
    a) Contiguous
    b) Random
    c) Both contagious and random
    d) None

Solution: (a) Contiguous

13. 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 11^{th} element which is not set, so it will print garbage value.

14. What will 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.

Such operations are not allowed. Wrong syntax used. Element wise addition has to be done in array using loop.

15. 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)

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