

## **Answer Script**

### **Question No. 01**

Update the following code so that the values in array b is 'x' more than the corresponding values in array a. You can only write only one line of code inside the loop (in the comment section marked "Write Code Here"). Do not modify anything else.

```
int n = 10;
int x = 5;
int index = 0;
int a[] = {12, 7, 3, 71, 2, 43, 38, 23, 45, 22};
int b[n];
for (int i=0; i<n; i++) {
    //Write Code Here
}
```

### **Answer No. 01**

```
int n = 10;
int x = 5;
int index = 0;
int a[] = {12, 7, 3, 71, 2, 43, 38, 23, 45, 22};
int b[n];
for (int i=0; i<n; i++) {
    //Write Code Here
    b[i] = a[i] + x;
}
```

### **Question No. 02**

```
char s = "america"
```

What is wrong with this C statement. Mark the errors and fix them

### **Answer No. 02**

In this C statement I find some errors,  
First error is not using the 3rd bracket after the variable to declare this string. We know the string is a character array, So when we declare an array we need to use 3rd brackets after the variable.

After this correction the C Statement is,

```
char s[] = "america"
```

Second error is not using a semicolon at the end of the statement. The semicolon tells that the current statement has been terminated.

After this correction the C statement is,

```
char s[] = "america";
```

### Question No. 03

Suppose you want to declare an array of size 15 and the elements of the array will be in a geometric progression, the first one starting with 1 and the common ratio being 2. For example, the first few elements of that array will be 1, 2, 4, 8, 16, ... and so on. Write a for loop to initialize the array with the required progression.

### Answer No. 03

```
int arr[15];  
arr[0] = 1;  
for(int i=1; i<15; i++){  
    arr[i] = arr[i-1] * 2;  
}
```

### Question No. 04

To declare a string, Fahim uses the following code. What will happen in the second and the third statement. Explain.

```
char a[10];  
a[0] = 98;  
a[1] = 97;  
a[2] = 'n';  
a[3] = 'a';  
a[4] = 'n';  
a[5] = 'a';  
a[6] = '\\0';
```

### Answer No. 04

The second statement and third statement are,

```
a[0] = 98;
```

```
a[1] = 97;
```

Here 98 an integer type data is assigned in a[0] character type data. Characters can not store the integer value in a char type, it will convert the integer value in the corresponding ASCII character and store it.

So, 98 ASCII Characters 'b' will assign in a[0],

97 ASCII Characters 'a' will be assigned in a[1].

### Question No. 05

Consider the following code.

```
char a[10];  
char ch = 'a';  
for(i = 0; i<8;i++)  
{  
    a[i] = ch+8-i;  
}  
a[8] = '\\0';
```

What will be stored in the given character array 'a' of the question after the execution of the block of code?

### Answer No. 05

"ihgfedcb" will store in an array after the code execution.

Inside the for loop, the statement assigns a character in the string index.

We know that in ASCII 'a' decimal is 97.

In the 1st iteration, i=0 then,

$97 + 8 - 0 = 105$ ,

So, the 105 ASCII character is 'i' and it is assigned in a[0].

In the 2nd iteration, i=1 then,

$97 + 8 - 1 = 104$ ,

So, the 104 ASCII character is 'h' and it is assigned in a[1].

In the 3rd iteration, i=2 then,

$97 + 8 - 2 = 103$ ,

So, the 103 ASCII character is 'g' and it is assigned in a[2].

In the 4th iteration,  $i=3$  then,

$$97 + 8 - 3 = 102,$$

So, the 102 ASCII character is 'f' and it is assigned in  $a[3]$ .

In the 5th iteration,  $i=4$  then,

$$97 + 8 - 4 = 101,$$

So, the 101 ASCII character is 'e' and it is assigned in  $a[4]$ .

In the 6th iteration,  $i=5$  then,

$$97 + 8 - 5 = 100,$$

So, the 100 ASCII character is 'd' and it is assigned in  $a[5]$ .

In the 7th iteration,  $i=6$  then,

$$97 + 8 - 6 = 99,$$

So, the 99 ASCII character is 'c' and it is assigned in  $a[6]$ .

In the 8th iteration,  $i=7$  then,

$$97 + 8 - 7 = 98,$$

So, the 98 ASCII character is 'b' and it is assigned in  $a[7]$ .

#### Question No. 06

How much bytes does an int type array with 12 elements take in memory?

#### Answer No. 06

We know that an int element takes 4 bytes in memory.

If,

1 element takes 4 bytes from memory

Then,

12 elements take  $(12 \times 4) = 48$  bytes from memory.

Ans: 48 bytes

#### Question No. 07

What does lexicographical comparison mean? Explain with examples. How can we order strings in lexicographic order in C?

### Answer No. 07

Part:1

A lexicographical comparison is the kind of comparison generally used to sort words alphabetically in dictionaries.

Part:2

Example:

Phitron ->Python-> Phero

Lexicographical Order: Phero->Phitron->Python

In lexicographical order, Phero comes before Phitron and Phitron comes before Python.

Part:3

Now we order the three strings in lexicographical order,

"abe", "abd", "abc"

We compare the two string by each character and check which are greater or less then we swap the string,

First we compare "abe" and "abd", where "abe" > "abd" because,  
a == a, b == b, e > d thats why "abe" > "abd".

Now, we swap this two string. After the swap the order right now,  
"abd", "abe", "abc".

We compare "abd" and "abc", where "abd" > "abc" because,  
a == a, b == b, d > c thats why "abd" > "abc".

Now we swap this two string. After the swap the order right now,  
"abc", "abe", "abd".

We compare "abe" and "abd", where "abe" > "abd" because,  
a == a, b == b, e > d thats why "abe" > "abd".

Now we swap this two string. After the swap the order right now,  
"abc", "abd", "abe". Which is sorted.

### Question No. 08

What does ASCII value of a character mean? How can you print the ASCII value of a character in C language?

### Answer No. 08

Part:1

In C programming, a character variable holds ASCII value (an integer number between 0 and 127) rather than that character itself. This integer value is the ASCII code of the character. For example, the ASCII value of 'B' is 66.

Part:2

We can print the ASCII values in C of any character by using the %d as a format specifier.

### Question No. 09

Explain Bubble Sorting. Suppose you are given an array of integers 12, 7, 9, 1, 3, 73, 11, 15, 62, 19, 4. What will be the sequence of these integers if we run Bubble sort for only 5 iterations?

### Answer No. 09

Bubble Sort:

Bubble sort is a basic algorithm for arranging a string of numbers or other elements in the correct order. Each pass in bubble sort, adjacent elements that are not in the correct order get swapped.

**Given array -**

**12, 7, 9, 1, 3, 73, 11, 15, 62, 19, 4**

**1st iteration -**

> 7, 12, 9, 1, 3, 73, 11, 15, 62, 19, 4

> 7, 9, 12, 1, 3, 73, 11, 15, 62, 19, 4

> 7, 9, 1, 12, 3, 73, 11, 15, 62, 19, 4

> 7, 9, 1, 3, 12, 73, 11, 15, 62, 19, 4

> 7, 9, 1, 3, 12, 11, 73, 15, 62, 19, 4

> 7, 9, 1, 3, 12, 11, 15, 73, 62, 19, 4

> 7, 9, 1, 3, 12, 11, 15, 62, 73, 19, 4

> 7, 9, 1, 3, 12, 11, 15, 62, 19, 73, 4

> 7, 9, 1, 3, 12, 11, 15, 62, 19, 4, 73

Total - 9 swaps done in 1st iteration and now the 1st largest value is in the sorted position.

**Present Array: 7, 9, 1, 3, 12, 11, 15, 62, 19, 4, 73**

**2nd iteration -**

> 7, 1, 9, 3, 12, 11, 15, 62, 19, 4, 73

> 7, 1, 3, 9, 12, 11, 15, 62, 19, 4, 73

> 7, 1, 3, 9, 11, 12, 15, 62, 19, 4, 73

> 7,1,3,9,11,12,15,19,62,4,73

> 7,1,3,9,11,12,15,19,4,62,73

Total - 5 swaps done in 2nd iteration and now the 2nd largest value is in the sorted position.

**Present Array: 7,1,3,9,11,12,15,19,4,62,73**

**3rd iteration -**

> 1,7,3,9,11,12,15,19,4,62,73

>1,3,7,9,11,12,15,19,4,62,73

>1,3,7,9,11,12,15,4,19,62,73

Total - 3 swaps done in 3rd iteration and now the 3rd largest value is in the sorted position.

**Present Array:1,3,7,9,11,12,15,4,19,62,73**

**4th iteration -**

> 1,3,7,9,11,12,4,15,19,62,73

Total - 1 swap done in 4th iteration and now the 4th largest value is in the sorted position.

**Present Array: 1,3,7,9,11,12,4,15,19,62,73**

**5th iteration -**

> 1,3,7,9,11,4,12,15,19,62,73

Total - 1 swap done in 5th iteration and now the 5th largest value is in the sorted position.

### Question No. 10

How can you find the sum of digits of a number? Write a C program that will extract the digits of the given integer number as input and add them to find the required output. For example, if the input is 1235623, then the output will be 22. Because,  $1+2+3+5+6+2+3 = 22$ .

### Answer No. 10

Part 1:

I will follow those steps to find the sum of digits of a number:

Step 1: Get the number by the user.

Step 2: Get the modulus/remainder of the number.

Step 3: sum the remainder of the number.

Step 4: Divide the number by 10.

Step 5: Repeat step 2 while the number is greater than 0.

Part 2:

```
#include <stdio.h>

int main() {
    int n, sum=0;
    scanf("%d", &n);
    while(n>0) {
        sum += n%10;
        n = n/10;
    }
    printf("%d", sum);
    return 0;
}
```

### Question No. 11

You will be given a 3x3 matrix as input. You need to check whether the matrix is a magic square or not. Magic squares are the matrices which have equal sum along all rows, columns and diagonals. For example,

2	7	6	→15	
9	5	1	→15	
4	3	8	→15	
↙15	↓15	↓15	↓15	↘15

This is a magic square as you can see all the sums are equal to 15 along all the rows and columns and even the diagonals.

### Answer No. 11

```
#include<stdio.h>
```



```

int main() {
    int size = 3;
    int matrix[3][3];
    int row,column = 0;
    int sum, sum1, sum2;
    int flag = 0;

    //Enter matrix :
    for (row = 0; row < size; row++) {
        for (column = 0; column < size; column++)
            scanf("%d", &matrix[row][column]);
    }

    //For diagonal elements
    sum = 0;
    for (row = 0; row < size; row++) {
        for (column = 0; column < size; column++) {
            if (row == column)
                sum = sum + matrix[row][column];
        }
    }

    //For Rows
    for (row = 0; row < size; row++) {
        sum1 = 0;
        for (column = 0; column < size; column++) {
            sum1 = sum1 + matrix[row][column];
        }
        if (sum == sum1)
            flag = 1;
        else {
            flag = 0;
            break;
        }
    }

    //For Columns
    for (row = 0; row < size; row++) {
        sum2 = 0;
        for (column = 0; column < size; column++) {

```

```
        sum2 = sum2 + matrix[column][row];
    }
    if (sum == sum2)
        flag = 1;
    else {
        flag = 0;
        break;
    }
}

if (flag == 1)
    printf("YES\n");
else
    printf("NO\n");

return 0;
}
```