

## Part I — Questions

1. Write a statement that creates a 10-elements `int` array named `numbers`.

**Solution.**

```
int[] numbers = new int[10];
```

2. In the following, what is the value of the size declarator? What is the value of the index?

```
int[] numbers;  
numbers = new int[8];  
numbers[4] = 9;
```

**Solution.** The size declarator is 8, the subscript, or index, is 4.

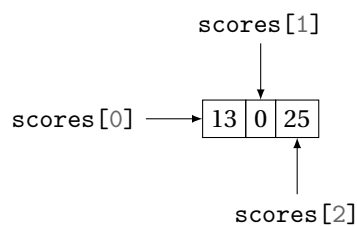
3. What is wrong with the following array declaration?

```
int[] books = new int[-1];
```

**Solution.** The size declarator cannot be negative.

4. Draw the content of the `scores` array once those statements have been executed.

```
int[] scores = new int[3];  
scores[0] = 13;  
scores[2] = 25;
```



**Solution.**

5. What will be displayed at the screen by the following program?

```
for (int num = 3 ; num <= 5 ; num++)  
    Console.WriteLine(num + " ");
```

**Solution.**

```
3 4 5
```

6. Write a `for` loop that display at the screen the sequence “1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ”.

**Solution.**

```
for (int x = 1 ; x <= 10 ; x ++)  
    Console.Write(x + ", ");
```

7. Adapt the code from Exercise 6 so that the same sequence will be displayed at the screen, but *without the last comma*.

**Solution.**

```
for (int x = 1 ; x <= 10 ; x ++)  
{  
    Console.Write(x);  
    if (x < 10) Console.Write(" ,");  
}
```

8. Write a for loop that display at the screen the sequence "1 3 5 7 9".

**Solution.**

```
for (int x = 1 ; x <= 10 ; x+= 2)  
    Console.Write(x + " ");
```

9. Consider the following code:

```
for (int y = 1; y <= 3; y++)  
{  
    for (int z = 1; z < 5; z++)  
        Console.Write("Scene " + y + ", take " + z + ". ");  
    Console.WriteLine();  
}
```

How many times does the outer loop iterates (i.e., how many scenes are shot)? How many times does the inner loop iterates (i.e., how many takes for each scene)? Finally, what is the total number of iteration of the nested loops (i.e., how many takes are made, total)?

**Solution.** 3, 4, 12.

10. Circle the pretest loops:

do while      switch      while      for      if-else-if

**Solution.** for and while are pretest loops.

11. What will be displayed at the screen by the following code?

```
int[] values = new int[6];  
for (int i = 0 ; i < 6 ; i++)  
    values[i] = (i*2);  
  
foreach (int j in values)  
    Console.WriteLine(j);
```

**Solution.**

```
0
2
4
6
8
10
```

12. Suppose we are given an `int` array `dailyPushUp` with 7 elements. Write a piece of code that display the value of the elements stored in the array `dailyPushUp`.

**Solution.**

```
for (int j = 0 ; j < 7 ; j++)
    Console.WriteLine(dailyPushUp[j]);
```

13. What is “array bounds checking”? When does it happen?

**Solution.** C# making sure that you’re not using a subscript outside the allowed range. It happens at run time.

14. Is there an error with the following code? If you think there is one, explain it, otherwise draw the content of the `myIncomes` array once those statements have been executed.

```
double[] myIncomes = new double[5];
myIncomes[1] = 3.5;
// No income on day two.
myIncomes[3] = 5.8;
myIncomes[4] = 0.5;
myIncomes[5] = 1.5;
```

**Solution.** The subscripts are off, they should go from 0 to 4.

15. What would be the size of the test array after the following statement has been executed?

```
int[] test = {3, 5, 7, 0, 9};
```

**Solution.** 5

16. Write a statement that creates and initializes a `double` array with the values 12.5, 89.0 and 3.24.

**Solution.**

```
double[] question = {12.5, 89.0, 3.24};
```

17. What is the value of `count` and the content of `number` once the following has been executed?

```
int count=2;
int[] number={3, 5, 7};
number[count--] = 8;
number[count]--;
```

**Solution.** count is 1. numbers is 3,4,8.

**18.** Suppose we have an array named `temp` that has been declared and initialized. How can we know the number of elements in this array?

**Solution.** By using the `Length` field: `temp.Length`.

**19.** Describe what the following code would do.

```
int[] record = { 3, 8, 11 };
int accumulator = 0;
foreach (int i in record)
    accumulator += i;
```

**Solution.** Declare and initialize an `int` array with the values 3, 8 and 11, and then sum those values in an accumulator variable.

**20.** Assuming we have two `int` arrays of the same size, `firstA` and `secondA`, write a program that copy the content of `firstA` into `secondA`.

**Solution.**

```
for (int k = 0 ; k < firstA.Length ; k++)
    secondA[k] = firstA[k];
```

**21.** Write a static method (header included) that takes as argument an `int` array, and display at the screen the value of each element of that array.

**Solution.**

```
public static void p(int[] a){
    foreach (int k in a) Console.WriteLine(k);
}
```

**22.** Write a static method (header included) that takes as argument an `int` array, and stores the value 10 in each element of that array.

**Solution.**

```
public static void z(int[] a){
    for (int j = 0 ; j < a.length ; j++) a[j]=10;
}
```

