

Programming Test Key

This test contains 10 questions for a total of 17 points.

1. (1 point) Which of the following data structure is a hierarchical data structure?
 - a. Tree**
 - b. Hash-map
 - c. Linked list
 - d. Array
 - e. None of the above
 - f. All of the above
2. (1 point) Which operation takes linear time in a double linked list:
 - a. Insertions at the start
 - b. Insertions at the end
 - c. Insertions at the middle**
 - d. All of the above
 - e. None of the above
3. (1 point) A double subscripted array declared as myArray[3][5] has how many elements?
 - a. 3
 - b. 5
 - c. 15**
 - d. 35
 - e. 10
 - f. Infinite amount.
4. (1 point) What is the random time access of an element in an array:
 - a. Linear
 - b. Log(n)
 - c. Quadratic
 - d. Constant**
 - e. None of the above
 - f. All of the above
5. (1 point) Which would be the best data structure to represent a relationship between keys and values:
 - a. Stack
 - b. Array
 - c. Map**
 - d. Tree
 - e. All of the above
 - f. None of the above
6. (1 point) Which data structure has on average constant time for lookup and insertion and the elements are unordered
 - a. Array
 - b. Linked Lists
 - c. Queue
 - d. Graph
 - e. Hash map**

7. (1 point) The benefit of Object Oriented Design is:

- a. Encapsulation
- b. Inheritance
- c. Polymorphism
- d. All of the above**
- e. None of the above

8. (2 points) What is the run time complexity of the following code:

```
sum = 0;
for ( i = 0; i < n; i++ ){
    for ( j = 0; j < n; j++){
        for ( k = 0; k < n; k++){
            sum += i+j;
        }
    }
}
```

- a) constant
- b) $O(n)$
- c) $O(n+n)$
- d) $O(n^2)$
- e) $O(j + i)$
- f) $O(n^3)$**

9. (2 points) What is output of the following code:

```
N = 10;
i = 1;
while ( i <= N )
{
    if ( i % 2 == 1 )// % is the modulo operator
    {
        print( i + " ");
    }
    i = i+1;
}
```

- a) 1 2 3 4 5 6 7 8 9 10
- b) i i i i i i i i i i i i i i i i i i
- c) 0 0 0 0 0
- d) 1 3 5 7 9**
- e) 2 4 6 8
- f) 2 4 6 8 10

10. (2 points) What is the output of **myFunction** when it is called with argument 3:

```
int myFunction(int arg)
{
    if ( arg <= 1 )
        return arg;
    else
        return arg * myFunction(arg - 1);
}
```

- a) 5 4 3 2 1
- b) 0
- c) 6
- d) 12
- e) 120
- f) 240

11. (4 points) Write a program in a high-level language or pseudocode for the following problems. You will not be graded on the syntax of your program, as long as the program description is clear.

- a. (2 points) Given an integer array M of size n, write a function that finds and returns the maximum element.

```
max = NULL;
if sizeof(M) != 0
{
    max = M(1);

    for m in M
        if m > max
            max = m;
}

return max;
```

2 points - They have to be in the general ball park, if they are comparing every element to a max variable and when the current variable is larger than the current max they assign the current variable to the max it should be fine.

1 point - they have a for loop, they are iterating over the elements but not assigning a max correctly.

- b. (2 points) Write a program that asks the user to enter a number bigger than 10. If the number is bigger than 10 the program prints the number the user entered multiplied by two and exits, otherwise it asks the user again, and keeps asking until a number bigger than 10 is entered.

```
answer = 0;
while ( answer <= 10 )
{
    answer = ask for a number bigger than 10;
    if ( answer > 10 )
    {
        print answer * 2;
        exit/break.
    }
}
```

2 points - same as before, they need to have a while loop and then an if statement inside checking that the condition is satisfied.

0.5 points only an if statement without a while loop