# Тест по Структури от данни и алгоритми Вариант 1

1. **Коя от следните структури от данни е най-подходяща за съхранение на последователност от символи?**

А) Стек

Б) Опашка

В) Списък

Г) Низ

1. **Коя структура от данни позволява да се запазват двойки от ключ и стойност и да се проверява дали даден ключ съществува?**

А) Стек

Б) Опашка

В) Списък

Г) Речник

1. **Каква е разликата между стек и опашка?**

А) Стекът е линейна структура от данни, която работи по принципа LIFO (last in, first out), а опашката е линейна структура от данни, която работи по принципа FIFO (first in, first out).

Б) Стекът е линейна структура от данни, която работи по принципа FIFO (first in, first out), а опашката е линейна структура от данни, която работи по принципа LIFO (last in, first out).

В) Стекът е нелинейна структура от данни, която работи по принципа LIFO (last in, first out), а опашката е нелинейна структура от данни, която работи по принципа FIFO (first in, first out).

Г) Няма разлика между стек и опашка.

1. **Каква е командата, която връща последния елемент на стек, без да го премахва от него?**

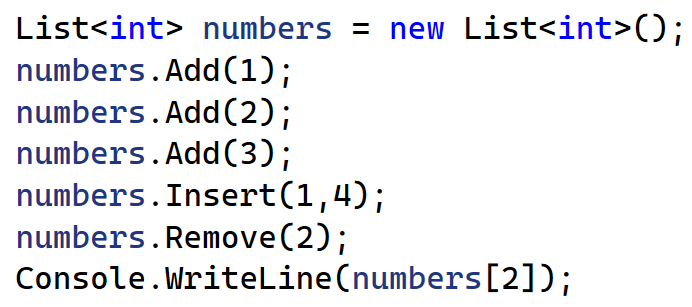
А) Push

Б) Pop

В) Peek

Г) Clear

1. **Отбележете какъв ще е изходът от следния програмен фрагмент:**

****

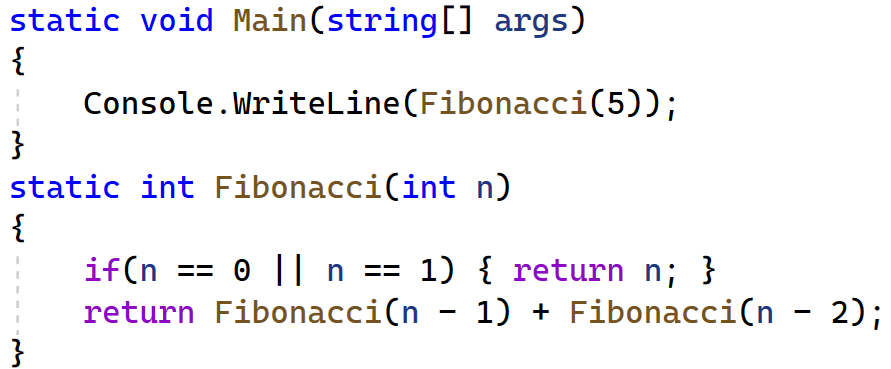
А) 1

Б) 2

В) 3

Г) 4

1. **Какво ще бъде изведено на конзолата след изпълнението на следния код?**

****

А) 0

Б) 1

В) 3

Г) 5

1. **В опашка последователно са постъпили елементи със стойност 14, 2, -3, 0, -2, 1, 9, а в стек: 4 и -4. От опашката се изважда елемент и ако той е положителен, се добавя в стека. След това стека премахва един елемент. Действието се повтаря до изваждането на елемент със стойност 0 от опашката.** 
   1. **Кои са елементите на стека?**

А) 0, -2, 1, 9  
Б) 14, 2, -3, 0  
В) 14, 2, -3  
Г) -2, 1, 9

* 1. **Кои са елементите на стека?**

А) -4, 1, -4  
Б) 4, -4   
В) -4, 1, -4, 9  
Г) 1, -4, -4, 9

1. **В дадения текст по-долу е обяснен алгоритъмът за двоично търсене. Попълнете липсващия текст.**

При двоично търсене в масив в началото се определят две променливи, които показват **(1)** и **(2)** индекс на масива. След това се намира средният индекс между тях и се сравнява елементът на този индекс с търсения елемент. Ако той е равен на търсения елемент, значи той е намерен. Ако той е по-голям от търсения елемент, значи търсеният елемент трябва да бъде в **(3)** половината на масива и се **(4)** крайният индекс на тази половина. Ако той е по-малък от търсения елемент, значи търсеният елемент трябва да бъде в **(5)** половината на масива и се **(6)** началният индекс на тази половина. Процесът се повтаря докато началният индекс не стане по-голям от крайния или да не се намери търсеният елемент.

(1) …

(2) …

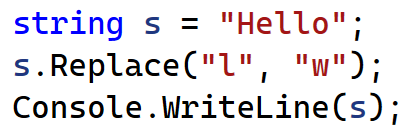
(3) …

(4) …

(5) …

(6) …

1. **Какво ще изведе следният програмен фрагмент?**

****

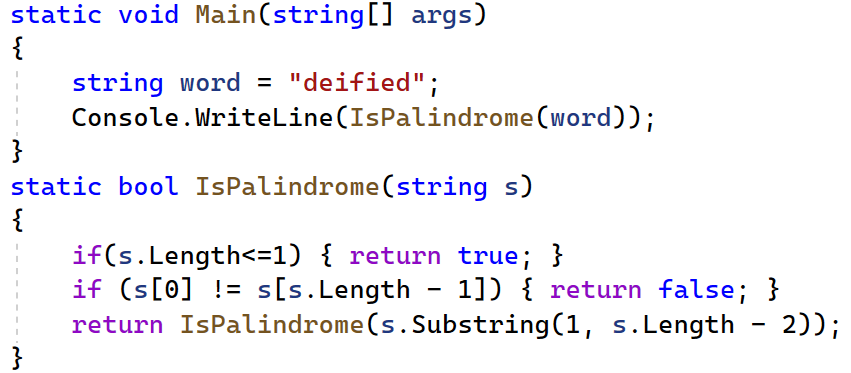
А) Hellow

Б) Hewwo

В) Hello

Г) Heowo

1. **Каква ще бъде изходната стойност на следния C# код?**

****

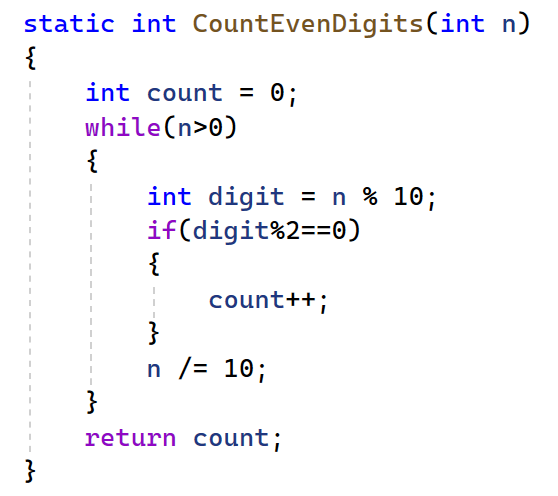
А) true

Б) false

В) deified

Г) error

1. **Даден е следният код. Ако n = 123456, какво ще върне рекурсията?**

****

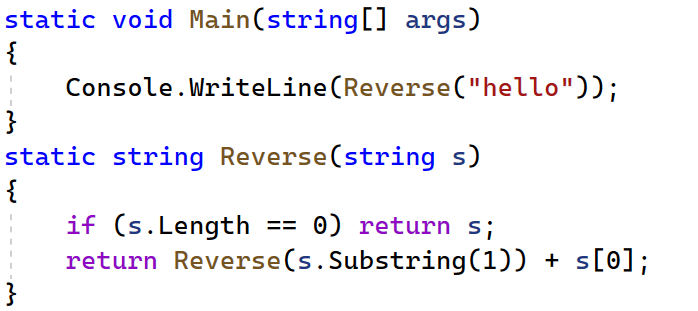
А) 0

Б) 2

В) 3

Г) 4

1. **Какво ще бъде изведено на конзолата след изпълнението на следния код?**

****

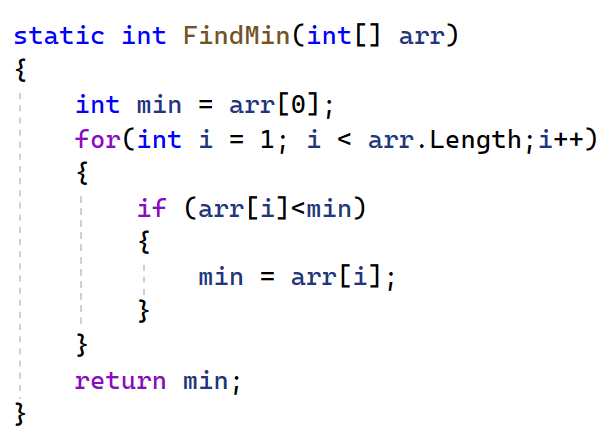
А) hello

Б) olleh

В) hleol

Г) lohel

1. **Даден ви е алгоритъм. Каква е неговата сложност?**

****

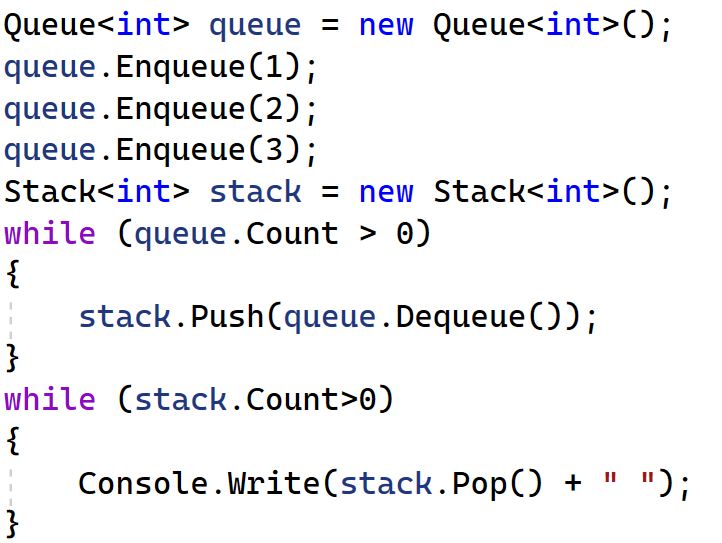
А) O(n)

Б) O(n log n)

В) O(n^2)

Г) (2^n)

1. **Какво ще се отпечата при изпълнение на следния C# код?**

****

А) 1 2 3

Б) 3 2 1

В) 1 3 2

Г) 2 3 1

**15.Кой от следните алгоритми за сортиране има най-ниска сложност в най-добрия си случай?**

А) Merge sort

Б) Quick sort

В) Bubble sort

Г) Insertion sort