

## Второ контролно по УП

### Вариант 1

**Задача 1.** Един масив ще наричаме **специален**, ако:

- Съдържа поне 3 елемента
- В него има специален елемент, за който:
  - Всички елементи от началото на масива до него включително са *сортирани строго възходящо*
  - Всички елементи от него включително до края на масива са *сортирани НЕстрого низходящо*

Напишете функция **getSpecialElement**, която приема указател към първия елемент на масив и неговия размер. Тя проверява дали един масив е **специален**. Ако е такъв, тя трябва да върне указател към специалния елемент в него, иначе да се върне **nullptr**.

Демонстрирайте работата на функцията в **main**, като въведете цяло положително число **n** (**n < 1024**) от клавиатурата и **n** на брой цели числа, представляващи елементите на масива. Изведете подходящо съобщение.

Вход	Изход	Пояснение на примера
5 1 3 5 5 0	The array is special, the special element is 5	Подмасивът 1 3 5 е сортиран строго възходящо Подмасивът 5 5 0 е сортиран нестрого низходящо
5 5 8 9 2 6	The array is not special	Подмасивът 9 2 6 не е сортиран низходящо

**Задача 2.** Напишете програма, която приема от стандартния вход естествено число **n** (**n < 1024**) и определя кои числа в интервала [1, ..., n] **не се делят и не са делители** на потребителските числа.

Потребителските числа се въвеждат едно по едно от стандартния вход (след въвеждането на **n**) и са **неограничен брой**. Въвеждането им приключва при подаване на числото 0.

Вход	Изход	Пояснение на примера
10 3 15 0	2 4 7 8 10	<ul style="list-style-type: none"><li>• 3, 6 и 9 се делят на 3</li><li>• 1 дели 3 и 15</li><li>• 5 дели 15</li></ul>
10 2 25 14 0	3 9	<ul style="list-style-type: none"><li>• 2, 4, 6, 8 и 10 се делят на 2</li><li>• 1 дели 2, 25 и 14</li><li>• 5 дели 25</li><li>• 7 дели 14</li></ul>

**Задача 3.** Дадена е матрица с  $n$  реда и  $m$  колони ( $n, m \leq 256$ ). Размерите на матрицата, както и нейните елементи се приемат от стандартния вход. Сортирайте **редовете** на матрицата по **произведението** на елементите на всеки ред във **възходящ ред**.

Вход	Изход	Произведение на всеки ред
4 4 1 2 3 4 -2 -5 10 1 100 50 0 3 4 -5 6 7	4 -5 6 7 100 50 0 3 1 2 3 4 -2 -5 10 1	-840 0 24 100

**Упътване:** Създайте си помощни функции:

`int getProductOf(const int matrix[][256], std::size_t columns, std::size_t rowIndex);` - намира произведението на ред #rowIndex

`void swap(int matrix[][256], std::size_t columns, std::size_t row1, std::size_t row2);` - разменя два реда от матрицата