

## Второ контролно по УП Вариант 2

**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 низходящо*
  - Всички елементи от него включително до края на масива са *сортирани НЕстрого възходящо*

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  **$n$  ( $n < 512$ )** от клавиатурата и  **$n$**  на брой цели числа, представляващи елементите на масива. Изведете подходящо съобщение.

Вход	Изход	Пояснение на примера
5 5 3 1 2 2	The array is special, the special element is 1	Подмасивът 5 3 1 е сортиран строго низходящо Подмасивът 1 2 2 е сортиран нестрого възходящо
5 3 2 0 7 2	The array is not special	Подмасивът 0 7 2 не е сортиран възходящо

**Задача 2.** Напишете програма, която приема от стандартния вход естествено число  **$n$  ( $n < 2048$ )** и определя кои числа в интервала  $[1, \dots, n]$  **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 <b>3</b></li><li>• 1 дели <b>3</b> и <b>15</b></li><li>• 5 дели <b>15</b></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 <b>2</b></li><li>• 1 дели <b>2</b>, <b>25</b> и <b>14</b></li><li>• 5 дели <b>25</b></li><li>• 7 дели <b>14</b></li></ul>

**Задача 3.** Дадена е матрица с  $n$  реда и  $m$  колони ( $n, m \leq 128$ ). Размерите на матрицата, както и нейните елементи се приемат от стандартния вход. Сортирайте **стълбовете** на матрицата по **сумата** на елементите на всеки ред в **низходящ ред**.

Вход	Изход	Сума на всеки стълб
4 4 1 -2 100 4 2 -5 50 -5 3 10 0 6 4 1 3 7	100 4 1 -2 50 -5 2 -5 0 6 3 10 3 7 4 1	153 12 10 4

**Упътване:** Създайте си помощни функции:

`int getSumOf(const int matrix[][128], std::size_t rows, std::size_t columnIndex);` - намира сумата на стълб #columnIndex

`void swap(int matrix[][128], std::size_t rows, std::size_t column1, std::size_t column2);` - разменя два стълба от матрицата