**Работа с масиви - сортиране**

Практически упражнения към курса [**"Programming Fundamentals" за ученици**](https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Fundamentals).

Тествайте задачите от тази тема в judge: <https://judge.softuni.bg/Contests/2655>

1. **Сравняване на символни масиви**

Сравняваме два масива лексикографски (буква по буква).

Извеждаме всеки по азбучен ред, всеки на нов ред

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| a b c  d e f | Abc  def |
| p e t e r  a n n i e | annie  peter |
| a n n i e  a n | An  annie |
| a b  a b | Ab  ab |

**Упътване**

* Сравняваме първите символи на **arr1[]** и **arr2[]**, ако са еднакви, сравняваме следващите и т.н.
* Ако всички символи са еднакви, по-малкия масив е по-късия
* Ако всички символи са еднакви и дължините им са равни, масивите са еднакви

1. **Вмъкване на елемент в сортиран масив**

Даден е сортиран масив от цели числа и друго цяло число. Напишете алгоритъм, който вмъква числото в масива, така, че масива отново да е подреден.

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 7 9  4 | 1 2 3 4 7 9 |
| 1 2 3 4 5  0 | 0 1 2 3 4 5 |

**Упътване**

* Сравняваме числото със средния елемент, ако е по-малко търсим в първата половина на масива (надолу), иначе – във втората (нагоре)
* Повтаряме горното правило докато масива, в който търсим има неповече от два елемента
* Мястото на числото е или преди по-малкия, или между двата или след по-големия елемент.
* Вмъкваме числото

1. **Търсене на елемент в сортиран масив**

Даден е сортиран масив от цели числа и друго цяло число. Напишете алгоритъм, който извежда “**Yes**” ако елемента се намира в масива и “**No**” ако елемента не се намира в масива.

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 7 9  7 | **Yes** |
| 1 2 3 4 5  0 | **No** |

**Упътване**

* Алгоритъмът е подобен на предишния

1. **Сливане на подредени масиви**

Създайте програма, която по зададени два подредени във възходящ ред масиви от цели числа, създава трети, който отново е подреден

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 2 3 7 9  2 4 5 7 8 | 1 2 2 3 4 5 7 7 8 9 |
| 1 2 3 4 5  1 7 9 10 | 1 1 2 3 4 5 7 9 10 |

**Упътване**

* Използвайте факта, че масивите са подредени
* Ако един елемент от единия масив е по-малък от елемент другия масив, то вземаме за последващата проверка елемент от същия масив иначе прилагаме правилото за другия масив

1. **Сортиране**

От клавиатурата се въвежда масив от цели числа. Сортирайте го в низходящ ред. Изведете резултата на един ред

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 1 5 -11 35 -3 | -11 -3 1 5 |
| 84 2 90 110 34 6 | 2 6 34 84 110 |

**Упътване**

* Намерете най-малкия елемент от масива и разменете мястото му с първия
* Повтаряйте горната стъпка с елементите от втория до последния и т.н

## Министерство на образованието и науката (МОН)

* Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".



* Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен** **лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

