

Работа с масиви - въвеждане и извеждане

Практически упражнения към курса "[Programming Fundamentals](#)" за ученици.

Тествайте задачите от тази тема в judge: <https://judge.softuni.bg/Contests/2636>

1. Статистика на масив

Напишете програма, която получава масив от цели числа (разделени с интервал) и извежда най-малкия елемент, най-големия елемент, сумата на елементите и средната им стойност.

Примери

| Вход | Изход | Вход | Изход |
|-------------|---|----------------------------|--|
| 2 3 4 5 6 1 | Min = 1 Max = 6 Sum = 21 Average = 3.5 | -1 200 124123 -400 -124214 | Min = -124214 Max = 124123 Sum = -292 Average = -58.4 |

Проверете решението си в judge системата.

Отворете страницата в judge за този урок: <https://judge.softuni.bg/Contests/2636>. Изберете задачата "Статистика на масив". Копирайте и поставете в тъмното поле **сорс кода**. Натиснете бутона за изпращане [Submit]:

→ ↺ ↻

judge.softuni.bg/Contests/Practice/Index/2636#0

Results

Submit a solution

Статистика на масив

Най-често срещано число

Индекс на буква

Преобразуване на масив в число

Обръщане на последователността на елементите

Обръщане на масив от символни низове

Завъртане и сумиране

Сгъни и събери

Статистика на масив

Условия

```
1 using System;
2 using System.Linq;
3
4 class ArrayStatistics
5 {
6     static void Main()
7     {
8         int[] numbers = Console.ReadLine().Split(' ').Select(int.Parse).ToArray();
9         //TODO:
10        Console.WriteLine($"Min = {numbers.Min()}");
11        Console.WriteLine($"Max = {numbers.Max()}");
12        Console.WriteLine($"Sum = {numbers.Sum()}");
13        Console.WriteLine($"Average = {numbers.Average()}");
14    }
15 }
```

Allowed working time: 0.100 sec.
Allowed memory: 16.00 MB
Size limit: 16.00 KB
Checker: Trim

C# code

Submit

Трябва да получите **100 точки** (напълно вярна задача):

| <div>⏮ ⏪ 1 ⏩ ⏭</div> | | | 🔄 |
|----------------------|----------------------------------|---------------------|---------|
| Points | Time and memory used | Submission date | |
| ✓✓✓✓✓ 100 / 100 | Memory: 8.66 MB Time: 0.046 s | 16:00:59 23.11.2020 | Details |

2. Най-често срещано число

Напишете програма, която намира най-често срещаното число в дадена последователност.

- Числата ще са в интервала [0...65535].
- В случай, че има няколко най-често срещани числа, изведете най-лявото от тях.

Примери

| Вход | Изход | Коментари |
|---------------------------|-------|--|
| 4 1 1 4 2 3 4 4 1 2 4 9 3 | 4 | Числото 4 е най-често срещаното (среща се 5 пъти) |
| 2 2 2 2 1 2 2 2 | 2 | Числото 2 е най-често срещаното (среща се 7 пъти) |
| 7 7 7 0 2 2 2 0 10 10 10 | 7 | Числата 2, 7 и 10 имат максимална честота (всяко се среща 3 пъти). Най-лявото е 7. |

3. Индекс на буква

Напишете програма, която създава масив, съдържащ всички букви от английската азбука (a-z). Въведете дума с малки букви (lowercase) от конзолата и изведете съответния индекс на всяка буква от масива с буквите от английската азбука.

Примери

| Вход | Изход |
|--------|---|
| abcz | a -> 0 b -> 1 c -> 2 z -> 25 |
| easter | e -> 4 a -> 0 s -> 18 t -> 19 e -> 4 r -> 17 |

4. Преобразуване на масив в число

Напишете програма, която въвежда масив от цели числа и го преобразува чрез сумиране на съседни двойки елементи, докато се получи едно цяло число. Например, ако имаме 3 елемента {2,10,3}, то събираме първите два и вторите два елемента и получаваме {2+10, 10+3} = {12, 13}, после събираме всички съседни елементи и получаваме obtain {12+13} = {25}.

Примери

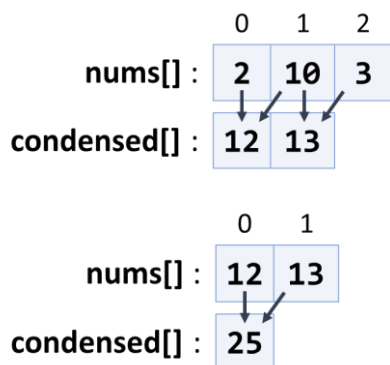
| Вход | Изход | Коментари |
|-----------|-------|--|
| 2 10 3 | 25 | 2 10 3 → 2+10 10+3 → 12 13 → 12 + 13 → 25 |
| 5 0 4 1 2 | 35 | 5 0 4 1 2 → 5+0 0+4 4+1 1+2 → 5 4 5 3 → 5+4 4+5 5+3 → 9 9 8 → 9+9 9+8 → 18 17 → 18+17 → 35 |
| 1 | 1 | 1 is already condensed to number |

Упътване

Докато имаме повече от един елемент в масива `nums[]`, повтаряй следното:

- Създай нов масив `condensed[]` с размер `nums.Length-1`.
- Събирай числата от `nums[]` в `condensed[]`:
 - `condensed[i] = nums[i] + nums[i+1]`
- `nums[] = condensed[]`

Процесът е илюстриран по-долу:



5. Обръщане на последователността на елементите на масив

Напишете програма, която въвежда масив от цели числа, **Обръща го** и извежда елементите. Входните данни са **числото n** (брой на елементите) + **n** цели числа, всяко на отделен ред. Изведете резултата на един ред, за разделител да се ползва интервал

Примери

| Вход | Изход |
|--------------------------|------------|
| 3 10 20 30 | 30 20 10 |
| 4 -1 20 99 5 | 5 99 20 -1 |

Упътване

- Първо, въведете числото n .
- Създайте масив от n цели числа.
- Въведете с цикъл `for` числата.
- Вместо да обръщате масива, можете просто да изведете елементите му като го обходите от последния до първия

6. Обръщане на масив от символни низове

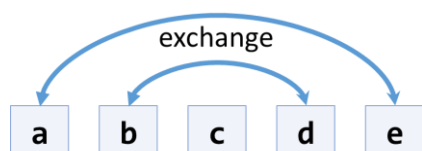
Напишете програма, която да прочете масив от символни низове, обръща масива и печата на неговите елементи. Входът се състои от поредица от низове, разделени с интервал. Отпечатва резултата на един ред с разделител интервал.

Примери

| Вход | Изход |
|------------|------------|
| a b c d e | e d c b a |
| -1 hi ho w | w ho hi -1 |

Упътване

- Въведете масив от символни низове
- Разменете първият елемент (с индекс 0) с последния елемент (с индекс $n-1$)
- Продължете с тези размени с останалите елементи докато стигнете средата на масива



- Друг, по-кратък подход е да се ползва готовия extension метод `.Reverse()` от `"System.Linq"`.

7. Завъртане и сумиране

“Завъртане на масив на дясно” означава да преместим неговия последен елемент на първо място: $\{1, 2, 3\} \rightarrow \{3, 1, 2\}$.

Напишете програма, която въвежда масив от n цели числа (разделени с интервал на един ред) и цяло число k , завърта k пъти надясно и сумира получените масиви след всяко завъртане както е показано по-долу:

Примери

| Вход | Изход | Коментари |
|---------------|---------|---|
| 3 2 4 -1 2 | 3 2 5 6 | rotated1[] = -1 3 2 4 rotated2[] = 4 -1 3 2 sum[] = 3 2 5 6 |
| 1 2 3 | 3 1 2 | rotated1[] = 3 1 2 |

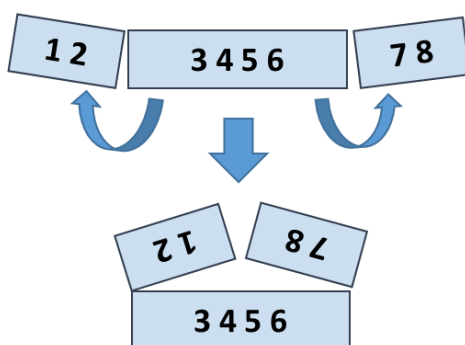
| | | |
|-----------|-------------|------------------------|
| 1 | | sum[] = 3 1 2 |
| 1 2 3 4 5 | 12 10 8 6 9 | rotated1[] = 5 1 2 3 4 |
| 3 | | rotated2[] = 4 5 1 2 3 |
| | | rotated3[] = 3 4 5 1 2 |
| | | sum[] = 12 10 8 6 9 |

Упътване

- След r завъртания, елементът на позиция i отива на позиция $(i + r) \% n$.
- Масивът `sum[]` може да бъде изчислен с два вложени цикъла : for $r = 1 \dots k$; for $i = 0 \dots n-1$.

8. Сгъни и събери

Въведете масив от $4 \times k$ цели числа, сгънете го както е показано по-долу и изведете сумата на горния и долния ред (всеки, съдържащ $2 \times k$ цели числа):



Примери

| Вход | Изход | Коментари |
|--|----------------|--|
| 5 2 3 6 | 7 9 | 5 6 + 2 3 = 7 9 |
| 1 2 3 4 5 6 7 8 | 5 5 13 13 | 2 1 8 7 + 3 4 5 6 = 5 5 13 13 |
| 4 3 -1 2 5 0 1 9 8 6 7 -2 | 1 8 4 -1 16 14 | -1 3 4 -2 7 6 + 2 5 0 1 9 8 = 1 8 4 -1 16 14 |

Упътване

- Създайте първия ред след сгъването: първите k числа **обърнати**, последвани от последните k числа, също **обърнати**.
- Създайте втория ред след сгъването, като вземете средните $2 \times k$ числа
- Сумирайте** първи и втори ред

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".



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- 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).



SoftUni
Foundation

