# Упражнения: Други типове данни

## Обръщане на знаци

Напишете програма, която въвежда **3 знака** и ги изведете в обратен ред.

**Примери**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| A  B  C | CBA |  | x  Y  z | zYx |  | G  g  n | ngG |

### Подсказки

За да въведете един символ, вземете само първия символ (този с индекс **0**) от прочетеното от конзолата:

**

## Изведете част от ASCII таблицата

Намерете в Интернет повече информация за [ASCII](http://www.ascii-code.com/) (American Standard Code for Information Interchange) и напишете програма, която извежда част от **ASCII таблицата** от знаци на конзолата. На първи ред на входа ще получите индекса на знака, от който трябва да започнете, а на втория ред – индекса на последния знак.

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 60  65 | < = > ? @ A |
| 69  79 | E F G H I J K L M N O |
| 97  104 | a b c d e f g h |
| 40  55 | ( ) \* + , - . / 0 1 2 3 4 5 6 7 |
|  |  |

### Подсказки

За да изведете например **латинската азбука**, можете да използвате **цикъл for**:



Може да се ползват и ASCII кодовете на символите:



Помислете как трябва да се промени горния програмен код, че да ви послужи в тази задача.

## Гласна или цифра

Напишете програма, която проверява дали даден символ е **цифра**, **гласна** или друг символ.

**Примери**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| A | vowel |  | 9 | digit |  | g | other |

## Поздрав

Напишете програма, която въвежда **първото име, последното име** и **възрастта** и извежда "***Hello, <first name> <last name>. You are <age> years old.***". Използвайте съставни низове.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Ivo  Hristov  23 | Hello, Ivo Hristov. You are 23 years old. |

### Подсказки

Може да използвате подобен код:



## Данни на служител

Компания иска да пази информация за служителите си. Всеки запис съдържа следната информация:

* Име
* Фамилия
* Възраст (0...100)
* Пол (m или f)
* ЕГН (e.g. 8306112507)
* Уникален номер на служителя (27560000…27569999)

Декларирайте променливите, необходими за да се пази информацията за един служител, като използвате подходящи типове данни. Използвайте описателни имена. Изведете данните на конзолата.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Amanda  Jonson  27  f  8306112507  27563571 | First name: Amanda  Last name: Jonson  Age: 27  Gender: f  Personal ID: 8306112507  Unique Employee number: 27563571 |

### Подсказки



## Създаване на дума

Напишете програма, която комбинира **n** знака и извежда на един ред **комбинацията** им

**Вход**

* На **първи ред**, ще получите **n** – броят на **редовете**, които ще **следват**
* На следващите **n реда** – ще получите **малки** и **големи** букви от **английската** азбука

**Изход**

**Изведете** на екрана **думата** във формата:

**The word is: {word}**

**Ограничения**

* **n** е в интервала **[1…20]**.
* Знаците винаги ще бъдат **букви от английската азбука**
* Ще получавате по една буква на ред

**Примери**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| **5**  A  b  C  d  E | The word is: AbCdE |  | **9**  C  o  d  e  R  u  l  z  z | The word is: CodeRulzz |

## Слепване на низове

Напишете програма, която въвежда **три** реда от конзолата. На **първи ред** ще въведете **разделител** (**char**) – трябва да **разделите** всички низове с този разделител. На **втори ред** ще получите “**even**” или “**odd**”. Ако получите “**odd**”, трябва да вземете всеки нечетен низ и обратното – ако получите “**even**”. На последния ред ще получите брой на редовете – **n**. Първата стъпка от цикъла започва от **1**.

Изведете получения низ на нов ред.

**Ограничения**

* **n** ще бъде в интервала **[1…20]**.
* Низовете ще бъдат не по-дълги от **30** знака

**Примери**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **изход** |  | **Вход** | **Изход** |
| -  **even**  **5**  One  Two  Three  Four  Five | Two-Four |  | **&**  **odd**  **4**  Pesho  Stefan  Maria  Gergana | Pesho&Maria |

**Подсказки**

* В C#, може да ползвате [String.Remove(…)](https://msdn.microsoft.com/en-us/library/d8d7z2kk(v=vs.110).aspx) за да премахнете последния разделител

## Низове и знаци

Напишете програма, която **задава знакови** и **низови стойности** на **променливи**. Бъдете сигурни, че всяка **стойност** се съхранява в **коректната променлива**. Накрая, трябва **да отпечатаме** всички променливи на конзолата.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| A string is sequence of chars  B  y  e  I love programming | A string is sequence of chars  B  y  e  I love programming |

### Подсказки

Декларирайте променливи от тип char или string, задайте им съответните стойности и ги **отпечатайте**.

## Асансьор

Изчислете колко курса ще трябва да направя един асансьор, за да се **качат n човека**, ако капацитета на асансьора е **p човека**. Входа се състои от два реда: **броя на хората n** и **капацитета p** на асансьора.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментар** |
| 17  3 | 6 | 5 курса \* 3 човека + 1 курс \* 2 човека |
| 4  5 | 1 | Всички хора се побират в асансьора.  Един курс е достатъчен. |
| 10  5 | 2 | 2 курса \* 5 човека |

### Подсказки

* Трябва да **разделите** n **на** p. Това дава броя на курсове с пълен капацитет (e.g. 17 / 3 = 5).
* Ако n не се дели точно на p, то ще трябва да се направи още един частично пълен курс (например 17 % 3 = 2).
* Друг начин е да се закръгли нагоре n / p към най-близкото цяло число (използвайки Math.Ceiling), например 17/3 = 5.67 🡪 се закръгля нагоре до 6.
* Примерен код за закръгляне:



## Сума на знаци

Напишете програма, която сумира ASCII кодовете на **n** знака и извежда сумата им.

**Вход**

* На **първи ред**, ще получите **n** – броят на **редовете**, които ще **следват**
* На следващите **n реда** – ще получите буквите от латинската азбука

**Изход**

Изведете общата сума в следния формат:

**The sum equals: {totalSum}**

**Ограничения**

* **n** е в интервала **[1…20]**.
* Знаците винаги ще бъдат малки или големи латински букви
* Ще се въвежда по една буква на ред

**Примери**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| **5**  A  b  C  d  E | The sum equals: 399 |  | **12**  S  o  f  t  U  N  i  R  u  l  z  z | The sum equals: 1263 |

## Декриптиране на съобщение

Ще получите **ключ** (**цяло число**) и **n** знака след това. Добавете ключа към всеки то знаците и ги слепете към съобщението. Накрая изведете полученото съобщение.

**Вход**

* На **първи ред**, ще получите **ключа**
* На **втори ред**, ще получите **n** – броя на редовете, които ще последват
* На следващите **n реда** – ще получите **малки** и **големи** букви от латинската азбука

**Изход**

Изведете декриптираното съобщение.

**Вход**

* **Ключът** ще бъде в интервала **[0…20]**
* **n** ще бъде в интервала **[1…20]**
* **Знаците** винаги ще бъдат малки или големи букви от латинската азбука
* Ще получавате по една буква на ред

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| **1**  **7**  C  d  b  q  x  o  s | Decrypt |

## \*Симулатор на лодка

Имате задача да направите симулатор на състезание с лодки. Ще получите две букви, които ще обозначават двете лодки.

След това ще получите **n случайни** низа. Всеки низ на нечетен ред показва скоростта на първата лодка, а на четен ред – скоростта на втората лодка. Лодката се предвижда с толкова позиции, колкото е дължината на съответния низ. Лодката, която първа стигне до **50**-та позиция е победител.

Лодките могат да се **ъпгрейдват**, което ще рече, че когато получим низа “**UPGRADE**” ще добавим 3 към **ASCII** кодовете и на двата знака на лодките и след това получените знаци ще се използват за визуализиране на лодките. Ако получите “**UPGRADE**”, лодките **НЕ се** мърдат.

Ако една от лодките стигне до **50** – изведете знака на победителя и спрете да приемате входни данни. Ако нито една от лодките не стигне до 50 – изведете тази, която е успяла да стигне до най-голямата позиция.

**Вход**

* На **първи ред**, ще получите знака на първата лодка
* На **втори ред**, ще получите знака на втората лодка
* На **трети ред,** ще получите **n** – броя на редовете, които ще последват

**Изход**

Изведете само знака на печелившата лодка.

**Ограничения**

* **n** ще е в интервала **[1…20]**
* Дължината на низовете ще е между **[1…100]** знака
* В края**,** лодките няма да имат еднакви позиции

**Примери**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментари** |
| **!**  **(**  **7**  move  need for speed  go  fast and furious  UPGRADE  stopTheBoat  UPGRADE | . | Първа лодка🡺 ‘**!**’, втора лодка 🡺 ‘**(**’  “**move**” 🡺 **4 знака** 🡺 **първа лодка (нечетен ред)** премества се с **4 позиции**  “**need for speed**” 🡺 14 знака 🡺 **втора лодка (четен ред)** премества се с **14 позиции**  “**go**” 🡺 2 знака 🡺 **първа лодка (нечетен ред)** премества се с **2** позиции.  “**fast and furious**” 🡺 16 знака 🡺 **втора лодка (четен ред)** премества се с **16** позиции.  “**UPGRADE**” 🡺 добавяме 3 към ‘**!**’ 🡺 става **‘$’**, добавяме 3 към **‘(‘** 🡺 става **‘+’**.  “**stopTheBoat**” 🡺 11 знака 🡺 **втора лодка (четен ред)** премества се **11** позиции.  “**UPGRADE**”🡺 добавяме 3 към ‘**$**’ 🡺 става **‘'’**, добавяме 3 към **‘+‘** 🡺 става **‘.’**.  **победител – втора лодка 🡺 41 позиция >** **6 позиция** 🡺 **втората лодка печели** |

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| **E**  **A**  **10**  UPGRADE  start  driveWithTheSpeedOfLight  go  driveWithTheSpeedOfLightOrFaster  Should not be read  a  Should not be read  b  Should not be read | H | Започваме с **UPGRADE** и първата лодка е представана с ‘**H**’, а втората с ‘**D**’  След **5ти** ред **първата** лодка е стигнала до **50** позиция и **НЕ** трябва да се приема вход от другите редове. |