# Упражнения: Булев тип данни

## Булева променлива

Напишете програма, която въвежда **низ**, преобразува го към променлива от **булев** тип и **извежда** “**Yes**”ако в променливата имаме true и “**No**” ако в променливата имаме false.

#### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| True | Yes |
| False | No |

#### Подсказки

* Използвайте [**Convert.ToBoolean(string)**](https://msdn.microsoft.com/en-us/library/86hw82a3(v=vs.110).aspx).

## Анализ на числа

Да се напише програма, която **чете цяло число**, въведени от потребителя, и проверява дали е **нула**, **положително**, **нечетно**, **валиден ден от месец**, **валиден месец**, **валиден час**, **валидна минута**. Да се изведе “true” или “false” като резултат на всяка от проверките.

#### Примери:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 6 | Zero: false  Positive: true  Odd: false  Month: true  Day: true  Hour: true  Minute: true | 0 | Zero: true  Positive: false  Odd: false  Month: false  Day: false  Hour: true  Minute: true | 25 | Zero: false  Positive: true  Odd: true  Month: false  Day: true  Hour: true  Minute: true |

#### Подсказка:

Декларирайте си целочислена променлива и прочетете в нея числото. Дефинирайте си булева променлива за всяка от проверките (например **IsZero**, **IsPositive** и т.н.) и укажете като нейна стойност какво трябва да се проверява за вече прочетеното число. После изведете стойността им.

## Еднакви думи

Да се напише програма, която **чете две думи**, въведени от потребителя, и проверява дали са еднакви. Да не се прави разлика между главни и малки думи. Да се изведе “yes” или “no”. Примери:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| Hello  Hello | yes | SoftUni  softuni | yes | Soft  Uni | no | beer  juice | no | HeLlO  hELLo | yes |

#### Подсказка:

Използвайте if-else конструкция. Преди сравняване на думите ги обърнете в долен регистър: word = word.ToLower().

1. **Познай паролата**

Да се напише програма, която **чете парола** (един ред с произволен текст), въведена от потребителя, и проверява дали въведеното **съвпада** с фразата “**s3cr3t!P@ssw0rd**”.При съвпадение да се изведе “**Welcome**”. При несъвпадение да се изведе “**Wrong password!**”. Примери:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| qwerty | Wrong password! | s3cr3t!P@ssw0rd | Welcome | s3cr3t!p@ss | Wrong password! |

#### Подсказка:

Използвайте **if**-**else** конструкцията.

1. **Точка в правоъгълник**

Напишете програма, която проверява дали **точка {x, y}** се намира **вътре в правоъгълник {x1, y1} – {x2, y2}**. Входните данни се четат от конзолата и се състоят от 6 реда, въведени от потребителя: десетичните числа **x1**, **y1**, **x2**, **y2**, **x** и **y** (като се гарантира, че **x1 < x2** и **y1 < y2**). Една точка е вътрешна за даден правоъгълник, ако се намира във вътрешността му или върху някоя от страните му. Отпечатайте “**Inside**” или “**Outside**”. Примери:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **визуализация** |  | **вход** | **изход** | **визуализация** |
| 2  -3  12  3  8  -1 | Inside |  | 2  -3  12  3  11  -3.5 | Outside |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **визуализация** |  | **вход** | **изход** | **визуализация** |
| -1  -3  4  1  0.5  1 | Inside |  | -1  -3  4  1  -1.2  1.4 | Outside |  |

#### Подсказка:

Една точка е вътрешна за даден многоъгълник, ако едновременно са изпълнени следните четири условия (можете да ги проверите с **if** проверка с логическо „**и**“ – оператор **&&**):

* Точката е надясно от лявата стена на правоъгълника (**x >= x1**)
* Точката е наляво от дясната стена на правоъгълника (**x <= x2**)
* Точката е надолу от горната стена на правоъгълника (**y >= y1**)
* Точката е нагоре от долната стена на правоъгълника (**y <= y2**)

1. **Плод или зеленчук**

Да се напише програма, която **чете име на продукт**, въведено от потребителя, и проверява дали е **плод** или **зеленчук**.

* Плодовете "**fruit**" са **banana**, **apple**, **kiwi**, **cherry**, **lemon** и **grapes**
* Зеленчуците "**vegetable**" са **tomato**, **cucumber**, **pepper** и **carrot**
* Всички останали са "**unknown**"

Да се изведе “**fruit**”, “**vegetable**” или “**unknown**” според въведения продукт. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| banana | fruit | apple | fruit | tomato | vegetable | water | unknown |

#### Подсказка:

Използвайте условна **if** проверка с логическо „**или**“ – operator **||**.

1. **Невалидно число**

Дадено **число е валидно**, ако е в диапазона [**100**…**200**] или е **0**. Да се напише програма, която **чете цяло число**, въведено от потребителя, и печата “**invalid**” ако въведеното число **не е валидно**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 75 | invalid | 150 | *(няма изход)* | 220 | invalid | 199 | *(няма изход)* |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| -1 | invalid | 100 | *(няма изход)* | 200 | *(няма изход)* | 0 | *(няма изход)* |

#### Подсказка:

Използвайте условна **if** проверка с **отрицание** и логически операции.

1. **Точка върху страната на правоъгълник**

Напишете програма, която проверява дали **точка {x, y}** се намира **върху някоя от страните на правоъгълник {x1, y1} – {x2, y2}**. Входните данни се четат от конзолата и се състоят от 6 реда въведени от потребителя: десетичните числа **x1**, **y1**, **x2**, **y2**, **x** и **y** (като се гарантира, че **x1 < x2** и **y1 < y2**). Да се отпечата “**Border**” (точката лежи на някоя от страните) или “**Inside / Outside**” (в противен случай). Примери:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** | **визуализация** |  | **вход** | **изход** | **визуализация** |
| 2  -3  12  3  8  -1 | Inside / Outside |  | 2  -3  12  3  12  -1 | Border |  |

#### Подсказка:

Използвайте една или няколко условни **if** проверки с логически операции. Точка **{x, y}** лежи върху някоя от страните на правоъгълник **{x1, y1} – {x2, y2}**, ако е изпълнено едно от следните условия:

* **x** съвпада с **x1** или **x2** и същевременно **y** е между **y1** и **y2**
* **y** съвпада с **y1** или **y2** и същевременно **x** е между **x1** и **x2**

Можете да проверите горните условия с една по-сложна **if**-**else** конструкция или с няколко по-прости проверки или с **вложени** **if**-**else** проверки.

1. **Клас животно**

Напишете програма, която отпечатва класа на животното според неговото име, въведено от потребителя.

* **dog, cat -> mammal**
* **crocodile, tortoise, snake -> reptile**
* **others -> unknown**

#### Примери:

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| dog | mammal |
| snake | reptile |
| eagle | unknown |

## \* Три еднакви числа

Да се напише програма, която чете 3 числа и отпечатва дали са еднакви (yes / no).