1 Declare Variables

Description

Declare five variables choosing for each of them the most appropriate of the types `byte`, `sbyte`, `short`, `ushort`, `int`, `uint`, `long`, `ulong` to represent the following values: `52130`, `-115`, `4825932`, `97`, `-10000`. Choose a large enough type for each number to ensure it will fit in it. Try to compile the code.

2 Float or Double

Description

Which of the following values can be assigned to a variable of type `float` and which to a variable of type `double`: `34.567839023`, `12.345`, `8923.1234857`, `3456.091`? Write a program to assign the numbers in variables and print them to ensure no precision is lost.

3 Variable in Hexadecimal Format

Description

Declare an integer variable and assign it with the value `254` in hexadecimal format (`0x##`). Use Windows Calculator to find its hexadecimal representation. Print the variable and ensure that the result is `254`.

4 Unicode Character

Description

Declare a character variable and assign it with the symbol that has Unicode code `42` (decimal) using the `\u00XX` syntax, and then print it.

Hint: first, use the Windows Calculator to find the hexadecimal representation of `42`. The output should be `*`.

5 Boolean Variable

Description

Declare a Boolean variable called `isFemale` and assign an appropriate value corresponding to your gender. Print it on the console.

6 Strings and Objects

Description

Declare two string variables and assign them with `Hello` and `World`. Declare an object variable and assign it with the concatenation of the first two variables (mind adding an interval between). Declare a third string variable and initialize it with the value of the object variable (you should perform type casting).

7 Quotes in Strings

Description

Write a program that outputs `The "use" of quotations causes difficulties.`

Input

- None

Output

- Print on a single line `The "use" of quotations causes difficulties.`

Constraints

- Time limit: 0.1s
- Memory limit: 16MB

8 Isosceles Triangle

Description

Write a program that prints an isosceles triangle of 9 copyright symbols `©`, something like this:

```
  ©

  © ©

 ©   ©

© © © ©
```

Note: The `©` symbol may be displayed incorrectly at the console so you may need to change the console character encoding to `UTF-8` and assign a Unicode-friendly font in the console.

Note: Under old versions of Windows the `©` symbol may still be displayed incorrectly, regardless of how much effort you put to fix it.

9 Exchange Variable Values

Description

Declare two integer variables a and b and assign them with `5` and `10` and after that exchange their values by using some programming logic. Print the variable values before and after the exchange.

10 Employee Data

Description

A marketing company wants to keep record of its employees. Each record would have the following characteristics:

- First name
- Last name
- Age (0...100)
- Gender (m or f)
- Personal ID number (e.g. 8306112507)
- Unique employee number (27560000…27569999)

Declare the variables needed to keep the information for a single employee using appropriate primitive data types. Use descriptive names. Print the data at the console.

11 Bank Account Data

Description

A bank account has a holder name (first name, middle name and last name), available amount of money (balance), bank name, IBAN, 3 credit card numbers associated with the account. Declare the variables needed to keep the information for a single bank account using the appropriate data types and descriptive names.

12 Null Values Arithmetic

Description

Create a program that assigns null values to an integer and to a double variable.

- Try to print these variables at the console.
- Try to add some number or the null literal to these variables and print the result.

13 Comparing Floats

Description

Write a program that safely compares two floating-point numbers (double) with precision `eps = 0.000001`.

Note: Two floating-point numbers `a` and `b` cannot be compared directly by `a == b` because of the nature of the floating-point arithmetic. Therefore, we assume two numbers are equal if they are more closely to each other than a fixed constant `eps`.

Input

- On the first line you will receive the first floating-point number
- On the second line you will receive the second floating-point number

Hint: Use double.Parse(Console.ReadLine()) to read input

Output

- Print `true` if the numbers are equal or `false` if they are not

Constraints

- Time limit: 0.1s
- Memory limit: 16MB

14 Print the ASCII Table

Find online more information about ASCII (American Standard Code for Information Interchange) and write a program that prints the visible characters of the ASCII table on the console (characters from `33` to `126` including).

Note: You may need to use for-loops (learn in Internet how).

Input

- None

Output

- The `94` characters on single line

Constraints

- Time limit: 0.1s
- Memory limit: 16MB