# Типове данни - реални типове

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/2648>

## Числа с плаваща запетая

Напишете програма, която **присвоява числа с плаваща запетая** на **променливи**. Уверете се, че всяка **стойност** се запазва в **коректен тип** (изберете най-удобния тип спрямо количеството памет, което той заема). Накрая трябва да изведете всички променливи.

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 3.141592653589793238  1.60217657  7.8184261974584555216535342341 | 3.141592653589793238  1.60217657  7.8184261974584555216535342341 |

## Лице на кръг (с точност 12 знака)

Напишете програма, в която въвеждаме радиус r (реално число) и **извеждаме лицето** на кръг с точно **12 знака** след десетичната запетая. Използвайте тип данни с **подходяща точност** за съхранение на резултатите.

### Примери

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| 2.5 | 19.634954084936 |  | 1.2 | 4.523893421169 |

### Подсказки

* Може да използвате тип double. Той има точност 15-16 знака.
* За да изведете точно 12 знака след десетичната запетая, може да ползвате следния код:



## Точна сума на реални числа

Напишете програма, която въвежда n числа и изчислява и извежда тяхната **точна сума** (без закръгляне).

### Примери

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| 3  1000000000000000000  5  10 | 1000000000000000015 |  | 2  0.00000000003  333333333333.3 | 333333333333.30000000003 |

### Подсказки

* Ако използвате типове като float или double, резултатът ще изгуби точността си. Също така данните може да се изведат с експоненциален запис. Може да използвате типа decimal, който съхранява реални числа с висока точност и по-малка загуба.
* Забележете, че decimal понякога съдържа ненужните нули след десетичната запетая, т.е. 0m е различно спрямо 0.0m и 0.00000m.

## Правоъгълник

Напишете програма, която изчислява за даден правоъгълник неговите **обиколка**, **лице** и **диагонал** по неговите страни.

### Примери

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |
| 10  5 | 30  50  11.1803398874989 |  | 22.1  10.2 | 64.6  225.42  24.3402958075698 |

### Подсказки

* Използвайте Math.Sqrt() за да изчислите диагонала (използвайте c2 = a2 + b2). Разгледайте <http://www.mathopenref.com/rectanglediagonals.html>.

## Преобразуване на скорост

Напишете програма, която въвежда **разстояние (в метри)** и време (като три числа: часове, минути, секунди), и **изведете скоростта**, в метри за секунда, километри в час и мили в час.

Приемете, че 1 миля = 1609 метра.

### Вход

* На първите ред ще получите – **разстояние в метри**
* На втория – **часове**
* На третия – **минути**
* На четвъртия – **секунди**

### Изход

Всяко число в изхода трябва да бъде изведено с точност 6 знака след запетаята

* На първи ред – скоростта в **метри в секунди** (m/s)
* На втори ред – скоростта в **километри в час** (km/h)
* На трети ред – скоростта в **мили в час** (mp/h)

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 1000  1  1  0 | 0.2732241  0.9836066  0.6113155 |  | 10000  0  20  30 | 8.130081  29.26829  18.19036 |  | 200000  2  5  0 | 26.66667  96  59.66439 |

### Подсказки

* Потърсете в интернет как да преобразувате мерните единици за скорост
* Типът float е достатъчно голям за тези изчисления.

## 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).

