
Zadanie 1

Napisz program rozwiązujący równanie kwadratowe ($ax^2 + bx + c = 0$) mając dane współczynniki a , b i c . Program powinien działać zawsze, również gdy wartości współczynników są zerowe lub wyróżnik jest ujemny.

Zadanie 2

Napisz program, który pyta użytkownika o jego/jej wzrost (w *metrach*, jako **double**) i wagę (w kilogramach, też jako **double**). Następnie program powinien wyświetlić jego/jej współczynnik BMI (*body mass index*) zdefiniowany jako waga w kilogramach podzielona przez kwadrat wzrostu w *metrach* – powinna to być liczba rzędu 20.

Zadanie 3

Napisz i przetestuj program, który

- Prosi użytkownika o wpisanie trzech liczb;
- Wczytuje te liczby;
- Następnie,
 - jeśli zdefiniowana jest nazwa (makro) preprocesora **GETMAX**, to program wyświetla na ekranie największą z wczytanych liczb;
 - jeśli zdefiniowana jest makro **GETMIN**, to program wyświetla na ekranie najmniejszą z nich;
 - jeśli zdefiniowane są obie nazwy, to wyświetla zarówno najmniejszą jak i największą z nich;
 - jeśli żadna z tych nazw nie jest zdefiniowana, to kompilacja powinna zostać przerwana.

Zadanie 4

Napisz program pobierający od użytkownika trzy nieujemne liczby całkowite i wypisujący na ekranie „histogram” dla tych danych, to znaczy trzy „słupki” złożone ze znaków `*`, wyrównane od dołu, o wysokościach równych wartościom trzech wczytanych liczb.

Wykonanie programu mogłoby zatem wyglądać tak:

```
$ ./histo
Enter three non-negative numbers: 3 1 8
*
*
*
*
*
```

```
* *  
* *  
***  
$
```

Nie używaj tablic, napisów ani żadnych innych kolekcji.

Zadanie 5

Napisz program, który w pętli prosi użytkownika o wpisanie dodatniej liczby całkowitej; wczytywanie kończy się, gdy użytkownik poda liczbę 0. Następnie program wypisuje tę z wczytanych liczb, dla której suma cyfr jest największa (oraz tę sumę cyfr).

Program komunikuje się z użytkownikiem w języku (np. polskim lub angielskim), który zależy od tego, czy zdefiniowane jest makro preprocesora (np. `POL` czy `ENG`); jeśli żadne z tych makr nie jest zdefiniowane, albo zdefiniowane są obydwa, to program nie powinien się w ogóle skompilować.

Przykładowy przebieg wykonania programu:

```
enter a natural number (0 if done): 23  
enter a natural number (0 if done): 59  
enter a natural number (0 if done): 78  
enter a natural number (0 if done): 91  
enter a natural number (0 if done): 0  
Max sum of digits was 15 for 78
```

UWAGA: Nie używać tablic, napisów ani żadnych innych kolekcji.

Zadanie 6

Napisać program grający z użytkownikiem w „20 pytań”. Użytkownik ustala (w myśli, może zapisać na karteczce) liczbę naturalną z przedziału $[1, 1\,000\,000]$. Program zadaje w pętli pytania *Czy jest to n ?*, na które użytkownik odpowiada wpisując literę

- *s* (jak w *small*) jeśli podana liczba n jest mniejsza od pomyślanej;
- *b* (jak w *big*) jeśli podana liczba n jest większa od pomyślanej;
- *y* (jak w *yes*) jeśli podana liczba n jest równa pomyślanej.

Program powinien kończyć się wyświetleniem komunikatu

Pomyślana liczba to ...

i podaniem prawidłowej odpowiedzi oraz liczby zadanych pytań. Nie używaj tablic, napisów ani żadnych innych kolekcji.
