

Zadanie 1 (10 pkt. na pracowni, później 5 pkt.). Napisz program, który dla podanego w parametrach wywołania programu ciągu znaków przedstawi efekt meksykańskiej fali (jeden cykl). Zasady:

- Załóż, że ciąg wejściowy składa się tylko z liter lub spacji, ale może być pusty.
- Program ma zademonstrować efekt fali, wypisując w kolejnych liniach (po jednej dla każdego znaku będącego literą) kolejne etapy procesu.
- Wielka litera reprezentuje osobę stojącą, mała – siedzącą.
- Jeśli znak w ciągu jest spacją, należy go pominąć w kolejnych transformacjach napisu tak, jakby było to puste miejsce (ale nadal należy go wydrukować).

Przykładowo, dla argumentu

Witaj Swiecie
należałoby wypisać

```
Witaj swiecie
wItaj swiecie
wiTaj swiecie
witAj swiecie
witaJ swiecie
witaj Swiecie
witaj sWiecie
witaj swIecie
witaj swiEcie
witaj swieCie
witaj swiecIe
witaj swieciE
```

a dla

ab CD
wyjście to

```
Ab cd
aB cd
ab Cd
ab cD
```

Uwaga: aby wprowadzić jako jeden argument wywołania napis zawierający spację, należy go otoczyć cudzysłowami. Zostaną one zinterpretowane przez powłokę (linię poleceń) i same nie zostaną przekazane jako część argumentu. Podobnie można przekazać pusty napis jako argument – wywołania dla powyższych przykładów oraz dla pustego napisu wyglądałyby odpowiednio

```
./a.out "Witaj Swiecie"
./a.out "ab CD"
./a.out ""
```

Zadanie 2 (10 pkt.). Napisz program, który dla zadanej w parametrach wywołania programu kwoty w złotych rozmieni ją na jak najmniejszą liczbę monet o nominałach: 5 zł, 2 zł, 1 zł, 50 gr, 20 gr, 10 gr, 5 gr, 2 gr, 1 gr. Program powinien wypisać "BLAD" w przypadku podania nieprawidłowego argumentu.

Uwaga: obliczenia należy wykonywać w groszach na liczbach całkowitych – typy zmiennopozycyjne niekoniecznie się do tego nadają (np. 0,1 to binarnie 0,0001100110011(0011)). Co za tym idzie, argument najlepiej przekonwertować na grosze samemu (lub używając funkcji bibliotecznych, ale nie przez proste, pojedyncze wywołanie `atof(argv[1])`). Należy uwzględnić również to, że separatorem dziesiętnym może być zarówno kropka, jak i przecinek.

Dla argumentu `10,75` wyjściem będzie

```
2x5zł
1x50gr
1x20gr
1x5gr
```

Argumenty takie jak `Ala`, `-5` czy `31.337` są błędne, ale `0` jest poprawny (wtedy wyjście będzie puste).

Zadanie 3 Podczas spaceru między grządkami kaktusów, gwatemalski matematyk zaczął rozważać następujący problem. Najpierw wyliczył wszystkie liczby *pierwsze* w przedziale $[n, m]$, które zawierają jednocześnie cyfry 3 i 7 (jak na przykład 13337, ale nie 3 czy 17). Następnie wybrał spośród tych liczb trzy różne x, y, z takie, że $x+y-z$ jest większe od pewnego ustalonego v , ale możliwie najmniejsze. Z powodu bycia rozproszonym przez kaktusy, matematyk nie jest jednak pewien swoich obliczeń w pamięci i potrzebuje potwierdzenia.

Na standardowym wejściu w pierwszym i jedynym wierszu podane są trzy liczby n, m, v takie, że $n, m, v \leq 2 \times 10^6$, $n < m$, $0 \leq v$.

Napisz program, który wczyta takie wejście i wypisze na standardowe wyjście jedną liczbę – minimalne $x+y-z$ większe od v , osiągalne dla dowolnie wybranych różnych liczb pierwszych x, y, z z przedziału $[n, m]$ i zawierających cyfry 3 i 7. Taka liczba wynikowa zawsze będzie istnieć dla podanych danych wejściowych.

Uwaga: warto zauważyć, że nie jest zbyt trudno wyliczyć żądane liczby pierwsze, jednak może być ich całkiem sporo, nawet kilkadziesiąt tysięcy.

Przykłady

A

Wejście:

```
1 200 20
```

Wyjście:

```
37
```

Matematyk wyliczył cztery liczby: 37, 73, 137, 173, tj. liczby pierwsze w przedziale [1,200] zawierające jednocześnie cyfry '3' i '7'. Mamy kombinację $73+137-173=37>20$, a na przykład $37+137-173=1$ jest za małe.

B

Wejście:

1 200 0

Wyjście:

1

$37+137-173=1$.

C

Wejście:

300 800 28

Wyjście:

31

Mamy $307+397-673=31$.