

Zadanie 1 (10 pkt. na pracowni, później 5 pkt.). Napisz program, który będzie w prosty sposób "szyfrował" dane wejściowe, przesuając cyklicznie samogłoski. Będziemy używać zapisanego na sztywno w kodzie parametru k , oznaczającego przesunięcia cykliczne w prawo w zbiorze samogłosek (czyli liter AEIOU). Przykładowo, 'e' przesunięte o 1 to 'i', 'O' przesunięte o 3 to 'E'.

Użyj $k=2$, ale ta wartość powinna być możliwa do łatwej modyfikacji przez zmianę pojedynczej zmiennej w kodzie.

Wskazówka: warto stworzyć dwie tablice, przechowujące samogłoski zapisane odpowiednio małymi i dużymi literami; można też użyć jednej tablicy, korzystając z faktu stałych odległości między dużymi/małymi literami w ASCII.

Tak jak na wykładzie, program powinien wczytywać wejście po jednym znaku, używając funkcji *getchar*, a wypisywać na standardowe wyjście przy pomocy funkcji *putchar*. Kiedy *getchar* zwróci *EOF*, program powinien zakończyć działanie. Jeśli wczytana jest samogłoska, należy wypisać ją na wyjście z przesunięciem k . Jeśli wczytany jest jakiegokolwiek inny symbol, powinien on zostać wypisany na wyjście bez zmian. Przykładowo, dla wejścia

```
Lepniak jest wielce miekki, konsystencja podobny do pieroga...
Oczy jego tepawe, wodniste. Lice gumiaste...
```

prawidłowym wyjściem byłoby

```
Lopnuik jost wuolco muokku, kansystoncji padabny da puoragi...
Aczy joga topiwo, wadnusto. Luco gemuisto...
```

Zadanie 2 (10 pkt.). Napisz program, który wczytuje ze standardowego wejścia liczbę $n \geq 3$ i następnie n liczb całkowitych. Nie używając sortowania (a najlepiej też bez tablic) wypisz 3 najmniejsze spośród tych liczb.

Następnie wypisz, która z tych trzech najmniejszych liczb ma najwięcej różnych nieparzystych czynników pierwszych w rozkładzie. Wyznaczanie liczby takich czynników powinno być realizowane w osobnej funkcji (np. wykorzystującej, z odpowiednimi modyfikacjami, kod z wykładu – *rozklad_dopierwiastka.c*).

Przykładowo, spośród liczb $16=2^4$, $21=3 \times 7$ i $125=5^3$, najwięcej różnych nieparzystych czynników pierwszych ma 21.

Zadanie 3. Meksykański biolog hoduje kaktusy na n grządkach po m kaktusów w każdej. Pierwszego dnia pożywka podawana jest kaktusom na pierwszej grządce, drugiego – na pierwszej i drugiej, k -tego – na grządkach od 1 do k , n -tego – na wszystkich grządkach, $n + 1$ -szego – znowu tylko na pierwszej, i tak dalej. Pod wpływem pożywki każdy kaktus (jeśli rośnie na odpowiedniej grządce) zmienia swoją wysokość z h na:

- $3h + 1$, jeśli h jest nieparzyste;
- $h / 2$, jeśli h jest parzyste.

Na standardowym wejściu w pierwszym wierszu podane są kolejno nieujemne $n \leq 150$, $m \leq 200$ oraz $d \leq 1000000$ – czas trwania eksperymentu w dniach. W kolejnych n wierszach

podanych jest po m nieujemnych liczb – wysokości kolejnych kaktusów na danej grządce, które nie przekraczają 38000000. Jeśli któraś wysokość jest podana jako 0, oznacza to, że na danym stanowisku nie ma żadnego kaktusa, więc taka wysokość ani się nie zmienia, ani nie wpływa na łączną wysokość wszystkich kaktusów.

Napisz program, który wczyta takie wejście i wypisze na standardowe wyjście jedną liczbę – największą osiągniętą w czasie trwania eksperymentu łączną wysokość kaktusów (uwzględniając zarówno stan sprzed pierwszego dnia podawania pożywki, jak i po d -tym dniu).

Przykłady

A

Wejście:

1 1 13

27

Wyjście:

322

Kaktus o wysokości 27 po czternastym dniu urosłby do wysokości 484 (a później jeszcze bardziej), ale tu eksperyment trwa tylko 13 dni.

B

Wejście:

2 2 1000

5 21

85 341

Wyjście:

1320

Takie kaktusy raz urosną, a potem będą się tylko kurczyć, ale pierwszego dnia rosną tylko niskie kaktusy z pierwszej grządki. Drugiego dnia rosną kaktusy z drugiej grządki (te z pierwszej już się kurczą) i to wtedy osiągnane jest maksimum (8+32 na pierwszej grządce, 256+1024 na drugiej).

C

Wejście:

3 3 0

80 756 764

93 44 198

179 0 113

Wyjście:

2227

$d = 0$, więc należy zwrócić sumę wysokości początkowych.

Uwagi i wskazówki

Zasady zmiany wysokości kaktusów nieprzypadkowo są takie, jak w hipotezie Collatza.

Podane ograniczenie wysokości pojedynczego kaktusa (38 milionów) dotyczy jej wartości początkowej.

Najlepiej używać wywołań *scanf* podobnych do *scanf("%d", &zmienna)*; (tj. zawierających w napisie formatującym tylko specyfikator(y) formatu, bez żadnych dodatków, w tym spacji) – wtedy można nie przejmować się tym, że na wejściu między liczbami czasami są spacje, a czasami znaki złamania wiersza; kolejne wywołania *scanf* po prostu wczytują kolejne liczby z wejścia, bez rozróżnienia, jakim dokładnie białym znakiem są rozdzielone.