

1 Opis problemu

Państwa zadaniem będzie złamanie prostego szyfru.

Szyfr ten jest następujący: każdy dopuszczalny znak z pliku wejściowego zastępujemy innym dopuszczalnym znakiem, wpisywanym do pliku wyjściowego.

Dopuszczalne w tym zadaniu znaki to **duże litery** alfabetu angielskiego oraz **średnik**(';') zastępujący w tekście kodowanym znaki przestankowe, ale kodowany jak każdy inny znak. W szczególności, nie ma **żadnych** białych znaków.

Taki szyfr jest stosunkowo prosty do złamania, trzeba tylko wcześniej zorientować się, z jakiego rodzaju szyfrem mamy do czynienia. Oprócz tego wystarczyłoby znać częstość występowania znaków kodowanych w wiadomości poddanej szyfrowaniu. Następnie:

1. Posortować znaki kodowane względem częstości występowania.
2. Posortować znaki z szyfrogramu względem częstości występowania.
3. Przyjąć, że znak kodujący o i-tej randze częstości występowania odpowiada znakowi kodowanemu (literze lub średnikowi) o i-tej randze częstości występowania - czyli najczęściej występujący znak odpowiada najczęściej występującej literze, drugi - drugiej itd.

Naturalnie, nie znamy częstości kodowanych znaków w samej szyfrowanej wiadomości, więc zakładamy, że są one zbliżone do częstości w języku szyfrowanej wiadomości (o ile go znamy) lub do częstości w pewnej bazie tekstów. Założenie takie może być błędne, zwłaszcza dla krótkich lub specyficznie konstruowanych szyfrogramów, dlatego w praktyce odszyfrowanie nawet tak prostych szyfrów wymaga trochę cierpliwości, a w przypadku łamania automatycznego - posiłkowania się słownikiem. W tym zadaniu przykłady testowe będą jednak tak dobrane, by powyższa prosta metoda dała poprawne rezultaty.

2 Wejście

Państwa program będzie przyjmował serie linii z poleceniami i ich parametrami. Po przyjęciu i odpowiedzi na jedną serię należy być gotowym do przyjęcia następnej (z wyjątkiem operacji **k**). Wejście dla jednego polecenia składa się z następujących linii:

1. Mała litera **o**, **u**, **w** lub **k** - kod operacji.

→ Jeśli kodem operacji było **u**:

2. Nazwa pliku tekstowego będącego źródłem wiedzy na temat częstości występowania znaków.
3. Nazwa pliku, do którego należy zapisać statystyki.

→ Jeśli kodem operacji było **o**:

2. Nazwa pliku tekstowego z zakodowanym tekstem.
3. Nazwa pliku z zapisanymi statystykami.
4. Nazwa pliku tekstowego, do którego należy wpisać odkodowany tekst.
5. Jeden znak, który będzie potrzebny przy wypisywaniu rezultatu.

→ Jeśli kodem operacji było **w**:

2. Nazwa pliku, z którego znaki będzie trzeba wyświetlić.
3. Numer pierwszego znaku, który trzeba wyświetlić (numerowane od 0) oraz liczbę znaków, które należy wyświetlić, oddzielone spacjami.

→ **k** nie ma argumentów, oznacza koniec programu.

3 Działanie programu i wyjście

→ Jeśli kodem operacji było `u` (uczenie), program:

1. Przeczyta plik z tekstem bazowym, “uczącym” (na podstawie którego robimy statystyki) i zliczy liczbę wystąpień wszystkich dopuszczalnych znaków oraz obliczy częstości.
2. Nadpisze plik ze statystkami w następujący sposób: dla każdego znaku, **którego liczba wystąpień jest większa od 0**, wpisze *do pliku ze statystkami* linię analogiczną do:

```
Znaleziono      168 znakow T, co stanowi 27.586% wszystkich.
```

Proszę zwrócić uwagę, że:

- (a) Na wpisanie liczby wystąpień zarezerwowane jest 8 znaków (licząc bez otaczających spacji). Jeżeli liczba nie zajmuje wszystkich ośmiu - należy dopełnić spacjami od lewej.
- (b) Częstość jest wyrażona w procentach, zawsze ma szerokość 6 (ew. dopełnione spacją po lewej) i dokładnie 3 miejsca po przecinku.
- (c) Spacje są tylko pomiędzy wyrazami i tylko po jednej (nie licząc spacji wynikających z punktu 2a lub 2b). Na końcu jest znak nowej linii.
- (d) Nie występują polskie znaki.

Linie te należy wypisać w porządku ustalonym przez częstości występowania liter - najpierw najczęściej występujące.

3. Na standardowe wyjście powinien wypisać tylko całkowitą liczbę zliczonych znaków.

→ Jeśli kodem operacji było `o` (odkodowanie), program:

1. Przeczyta plik z szyfrogramem, podany w linii 2, zliczy występowanie znaków i wypisze na *standardowe wyjście* linie analogiczne do tych z punktu 2 powyżej.
2. Wczyta statystyki z pliku sformatowanego, jak powyżej, podanego w linii 3.
3. Dopasuje znaki kodujące do kodowanych zgodnie z algorytmem przedstawionym w sekcji “Opis problemu”. Wypisze na `stdout`, dla wszystkich znaków kodujących, **dla których statystyki (z pliku z szyfrogramem) są większe od 0**, linie następującego typu:

```
Symbol A koduje U
```

Spacje są tylko pomiędzy wyrazami i tylko po jednej. Na końcu jest znak nowej linii.

Sortowanie jest alfabetyczne, wg znaku kodującego (średnik na końcu)

W testach nie występują równe niezerowe częstości znaków.

4. Odkoduje plik z linii 2 i **dopisze** wynik do pliku z linii 4.
5. Wypisze na `stdout` 50 znaków od **drugiego od końca** wystąpienia znaku podanego w linii 5 w **pliku** podanym w linii 4. Jeśli od tego wystąpienia jest mniej niż 50 znaków do końca, to oczywiście wypisujemy tylko tyle znaków, ile jest (nie dopełniamy spacjami itp.). Można założyć, że podany znak wystąpi w tekście przynajmniej dwa razy.

→ Jeśli kodem operacji było `w`: z pliku podanego w linii 2 program pominie tyle znaków ile wynosiła pierwsza liczba z linii 3, a następnie wypisze na standardowe wyjście, tyle ile wynosiła druga liczba z linii 3, chyba że wcześniej napotka koniec pliku. Podany plik może zawierać dowolne znaki ASCII (bez kontrolnych).

→ Jeśli kodem operacji było `k`, należy zakończyć program.

Należy przesłać plik `cpp`, nie `spakowany`

Białe znaki w odpowiedziach **będą** sprawdzane.

Proszę **nie** wczytywać dużych części plików do pamięci - będzie jej dostępne tak mało, że niektóre pliki po prostu się nie zmieszczą

Błąd w jednej z podanych komend powoduje od razu błąd danego testu.

Wiele operacji powtarza się w różnych kontekstach (np. wypisywanie linii ze statystyką, sortowanie ze względu na częstość wystąpień, liczenie częstości znaków w pliku). Dobrze byłoby to wykorzystać.

4 Przykład

Dla pliku "uczającego": TAKTYK;KATATYKA;KASTYSTATYSTA uprzednio napisanego pliku z szyfrogramem: OMNOM i szyfrogramu UHOWUHU;WU;OHIUHO;OH, po wywołaniu i wpisaniu :

```
u
jawny_ucz.txt
jawny_stat.txt
w
jawny_stat.txt
0 1000
w
jawny_stat.txt
45 88
o
jawny_kod.txt
jawny_stat.txt
jawny_dekod.txt
K
w
jawny_dekod.txt
0 100
k
```

Plik ze statystykami jawny_stat.txt powinien zawierać:

```
Znaleziono      8 znakow T, co stanowi 27.586% wszystkich.
Znaleziono      7 znakow A, co stanowi 24.138% wszystkich.
Znaleziono      5 znakow K, co stanowi 17.241% wszystkich.
Znaleziono      4 znakow Y, co stanowi 13.793% wszystkich.
Znaleziono      3 znakow S, co stanowi 10.345% wszystkich.
Znaleziono      2 znakow ;, co stanowi  6.897% wszystkich.
```

Na końcu jest nowa linia. Plik odkodowany powinien zawierać:

```
OMNOMTAKSTATYSTYKA;TAKTYKA
```

Na standardowe wyjście powinno zostać wypisane:

```
29Znaleziono      8 znakow T, co stanowi 27.586% wszystkich.
Znaleziono      7 znakow A, co stanowi 24.138% wszystkich.
Znaleziono      5 znakow K, co stanowi 17.241% wszystkich.
Znaleziono      4 znakow Y, co stanowi 13.793% wszystkich.
Znaleziono      3 znakow S, co stanowi 10.345% wszystkich.
Znaleziono      2 znakow ;, co stanowi  6.897% wszystkich.
86% wszystkich.
Znaleziono      7 znakow A, co stanowi 24.138% wszystkich.
Znaleziono Znaleziono      6 znakow U, co stanowi 28.571% wszystkich.
Znaleziono      5 znakow H, co stanowi 23.810% wszystkich.
Znaleziono      4 znakow O, co stanowi 19.048% wszystkich.
Znaleziono      3 znakow ;, co stanowi 14.286% wszystkich.
Znaleziono      2 znakow W, co stanowi  9.524% wszystkich.
Znaleziono      1 znakow I, co stanowi  4.762% wszystkich.
Symbol H koduje A
Symbol I koduje ;
Symbol O koduje K
Symbol U koduje T
Symbol W koduje S
Symbol ; koduje Y
KTYKAOMNOMTAKSTATYSTYKA;TAKTYKA
```

Niektóre linie wydają się sklejone, ponieważ operacje zasadniczo nie wypisują znaku nowej linii na końcu.

Jeżeli zamiast K jako ostatni argument operacji o podanoby M, to w ostatniej linii powinno się znaleźć:

```
MNOMTAKSTATYSTYKA;TAKTYKAOMNOMTAKSTATYSTYKA;TAKTYKA
```