Zadanie: SZY
Szyfry

Olimpiada Informatyczna

ONTAK 2013, dzień 6. Plik źródłowy `szy.*` Dostępna pamięć: 64 MB. *12.08.2013*

Bajtoccy szpiedzy przechwycili pięć zaszyfrowanych wiadomości. Agencja wywiadowcza BIA dowiedziała się, jakimi algorytmami są one zaszyfrowane, oraz dostała informację, że teksty:

- są w języku angielskim,
- zawierają tylko znaki o kodach ASCII od 32 do 126, oraz znak nowego wiersza o kodzie 10,
- nie zawierają małych liter.

Twoim zadaniem jest rozszyfrować przechwycone wiadomości. Za to zadanie można zdobyć co najwyżej 180 punktów.

Wejście

W dziale Pliki znajdziesz paczkę z załącznikiem do tego zadania. Znajdują się w niej *wszystkie* wejścia, na których oceniane będzie Twoje rozwiązanie. Każdy plik wejściowy zawiera zaszyfrowaną wiadomość. Dokładna specyfikacja wejścia podana jest w opisach kolejnych podzadań.

Wyjście

Na wyjście należy wypisać odszyfrowaną wiadomość.

Wiadomość 0 (0 punktów)

Ta wiadomość z przykładu. Jest ona zaszyfrowana przez odwrócenie kolejności bajtów wiadomości.

Pierwszy wiersz wejścia zawiera słowo `reverse`. Drugi wiersz zawiera liczbę oznaczającą długość tekstu. Trzeci wiersz zawiera kody ASCII kolejnych znaków tekstu, w odwróconej kolejności.

Wiadomość 1 (30 punktów)

Ta wiadomość zaszyfrowana jest szyfrem Cezara. Każda wielka litera zamieniona jest na literę znajdującą się o k pozycji dalej w porządku alfabetycznym (dla pewnego ustalonego k). Rozważamy alfabet angielski i zakładamy, że kolejnymi literami po literze `Z` są `A`, `B`, `C`, ...

Przykładowo, jeśli $k = 2$, to `A` zamienione jest na `C`, a `Z` na `B`. Wszystkie znaki nie będące wielkimi literami pozostawione są bez zmian.

Pierwszy wiersz wejścia zawiera słowo `caesar`. Drugi wiersz zawiera liczbę oznaczającą długość tekstu. Trzeci wiersz zawiera kody ASCII kolejnych znaków zaszyfrowanego tekstu.

Wiadomość 2 (50 punktów)

Ta wiadomość zaszyfrowana jest szyfrem XOR zdefiniowanym poniżej.

Klucz składa z k ($1 \leq k \leq 30$) bajtów $b_0, \ldots, b_{k-1}$. i-ty bajt tekstu t_i ($i \geq 0$) zaszyfrowany jest przy pomocy wzoru t_i xor $b_{i \bmod k}$.

Pierwszy wiersz wejścia zawiera słowo `xor`. Drugi wiersz zawiera liczbę określającą długość tekstu. Trzeci wiersz zawiera kody ASCII kolejnych znaków zaszyfrowanego tekstu.

Wiadomość 3 (50 punktów)

Ta wiadomość zaszyfrowana jest szyfrem podstawieniowym. Każdą wielką literę $l \in \{\texttt{A}, \ldots, \texttt{Z}\}$ zamieniamy na literę $f(l) \in \{\texttt{A}, \ldots, \texttt{Z}\}$. Żadne dwie litery nie są zamieniane na tę samą literę. Innymi słowy f to permutacja wielkich liter. Inne znaki pozostawione są bez zmian.

Pierwszy wiersz wejścia zawiera słowo `substitution`. Drugi wiersz zawiera liczbę określającą długość tekstu. Trzeci wiersz zawiera kody ASCII kolejnych znaków zaszyfrowanego tekstu.

Wiadomość 4 (50 punktów)

Ta wiadomość zaszyfrowana jest szyfrem RSA.

Pierwszy wiersz wejścia zawiera słowo `rsa`. Drugi wiersz wejścia zawiera liczbę n. Trzeci wiersz wejścia zawiera liczbę e. Czwarty wiersz wejścia zawiera liczbę E, reprezentującą zaszyfrowaną wiadomość.

Liczby n i e to klucz publiczny szyfru RSA. Liczba n jest postaci $n = pq$, gdzie p i q to dwie liczby pierwsze. Ponadto $p, q \equiv 2 \pmod 3$ oraz $e = 3$.

Wiadomość najpierw zamieniamy na liczbę M, przez potraktowanie tekstu jako ciągu cyfr w systemie o podstawie 27. Pierwszy znak tekstu odpowiada najbardziej znaczącej cyfrze. Literę A odpowiada wartości 1, B wartości 2 i tak dalej. Wszystkie znaki wiadomości to wielkie litery.

Zaszyfrowana wiadomość to liczba $E = M^e \bmod n$, gdzie $M < n$. Należy odszyfrować wiadomość, skonwertować do postaci tekstowej i wypisać na wyjście.

Wskazówka pierwsza Niech p będzie liczbą *pierwszą* oraz a będzie liczbą względnie pierwszą z p. Wówczas $a^{p-1} \equiv 1 \pmod p$.

Wskazówka druga Na Twoim komputerze zainstalowany jest program `bc`, który potrafi obliczać wartości prostych wyrażeń arytmetycznych i obsługuje liczby o wielu dziesiątkach cyfr.

Przykład

Dla danych wejściowych:

```
reverse
14
69 71 65 83 83 69 77 32 84 69 82 67 69 83
```

poprawnym wynikiem jest:

```
SECRET MESSAGE
```