

# Zadanie: MON

## Pięć monet



III OI, etap drugi, dzień próbny. Plik źródłowy `mon.*` Dostępna pamięć: 256 MB.

Mamy pięć monet  $A, B, C, D$  oraz  $E$ , których ciężary są parami różne. Dysponując bardzo czułą wagą szalkową, pozwalającą porównać ze sobą ciężary dwóch monet, uszereguj je od najlżejszej do najcięższej. Waga jest delikatnym i kosztownym urządzeniem. Pierwszych siedem porównań możemy wykonać za darmo. Ósme kosztuje 1 złoty, a każde następne porównanie jest 5 razy droższe od poprzedniego. Decyzję o tym, które porównanie wykonać w danej chwili, możemy podejmować w zależności od wyników poprzednich pomiarów. Zaprojektuj algorytm sortujący monety, tak aby koszt ważenia był jak najmniejszy.

## Komunikacja

Twoim zadaniem jest napisanie procedury `sortujMonety`. Ta procedura powinna, korzystając z funkcji `lzejsza`, uszeregować monety według ich ciężaru od najlżejszej do najcięższej, a następnie, wywołując procedurę `wynik`, przekazać znalezioną kolejność. Procedura ta nie powinna czytać nic ze standardowego wejścia, ani wypisywać nic na standardowe wyjście (może natomiast używać standardowego strumienia błędów).

Ocena poprawności rozwiązania dla pewnych dowolnie wybranych ciężarów monet zostanie dokonana na podstawie tego wywołania procedury `wynik`. Zwróć uwagę, że procedura `sortujMonety` może zostać wywołana kilkakrotnie w jednym biegu programu. Funkcja `lzejsza` oraz procedura `wynik` są dostępne w języku C++ przez plik nagłówkowy `mon.h` i mają następujące sygnatury:

- `bool lzejsza(char x, char y)` – funkcja ta zwraca `true`, jeżeli moneta o znaku  $x$  jest lżejsza od monety  $y$ , bądź `false` w przeciwnym wypadku. Monety te powinny być różne i ich znaki powinny należeć do zbioru  $\{A, B, C, D, E\}$ .
- `void wynik(char p, char q, char r, char s, char t)` – procedura ta powinna zostać wykorzystana do odpowiedzi na dane zapytanie. Monety powinny być uszeregowane w kolejności od najlżejszej do najcięższej ( $p$  jest najlżejszą monetą, drugą najlżejszą winna być  $q$ , itd.).

Twoim zadaniem jest napisanie procedury `void sortujMonety()`.

## Przykład

Przykładowe wywołanie procedury `sortujMonety()` może wyglądać następująco:

wywołanie	wynik
<code>lzejsza('A', 'B')</code>	<code>true</code>
<code>lzejsza('C', 'B')</code>	<code>false</code>
<code>lzejsza('C', 'D')</code>	<code>true</code>
...	
<code>lzejsza('E', 'D')</code>	<code>true</code>
<code>wynik('A', 'B', 'C', 'D', 'E')</code>	

## Ocenianie

Niech  $T$  oznacza liczbę przypadków testowych, a  $C$  łączny koszt użycia wagi we wszystkich przypadkach testowych. Wtedy Twój program otrzyma  $\max(0, \lfloor 100 - \frac{80C}{25T} \rfloor)$  punktów.