

# Zadanie: ARC

## Architekci



XVI OI, etap II. Plik źródłowy arc.\* Dostępna pamięć: 32 MB.

Król Bajtazar postanowił wybudować sobie nowy pałac. Ogłosił więc konkurs na najlepszy projekt architektoniczny pałacu. Chcąc zmotywować architektów do spiesznej pracy, ogłosił też, że projekty będzie rozpatrywał w takiej kolejności, w jakiej będą nadsyłane.

Z tym zleceniem wiąże się ogromny prestiż, dlatego też architekci z całego świata nadsyłają swoje propozycje do królewskiej kancelarii. Projektów nadchodzi bardzo dużo, a Bajtazar nie ma czasu ich wszystkich przeglądać. Zrezygnował zatem z samodzielnego wykonywania tej czynności i poprosił swojego kanclerza o to, by wstępnie przejrzał nadchodzące projekty zgodnie z następującymi zasadami:

- kanclerz ma wybrać  $k$  projektów, odrzucając resztę od razu - Bajtazar wie, że więcej niż  $k$  projektów i tak nie będzie w stanie przejrzeć;
- projekty mają zostać przedstawione Bajtazarowi w takiej kolejności, w jakiej zostały nadesłane - w takiej też kolejności Bajtazar będzie je przeglądał, zgodnie z tym co ogłosił;
- spośród wszystkich ciągów  $k$  projektów spełniających powyższe warunki, kanclerz ma wybrać ciąg **najlepszy**, zgodnie z poniższą definicją:  
Powiemy, że ciąg projektów  $(p_1, p_2, \dots, p_k)$  jest **lepszy** od ciągu  $(r_1, r_2, \dots, r_k)$ , jeśli dla pewnego  $l \geq 1$  pierwszych  $l - 1$  projektów w obu ciągach jest równie dobrych, zaś  $l$ -ty projekt w ciągu  $p$  jest lepszy od  $l$ -tego projektu w ciągu  $r$  (czyli  $p_i = r_i$  dla  $i < l$  i  $p_l > r_l$ ).

Projekty cały czas nadchodzą i nie wiadomo, do kiedy Bajtazar rozkaże je przyjmować. Kanclerz nie chce zostawiać wyboru  $k$  projektów na ostatni moment, jednak bardzo boi się popełnić błąd i narazić na gniew króla. Dlatego poprosił Cię o pomoc.

## Komunikacja

Twój program nie powinien nic czytać z wejścia ani wypisywać na wyjście. Zamiast tego zostanie on skompilowany z biblioteką oceniającą. Na początku swojego programu umieść dyrektywę

```
#include "arc.h"
```

Biblioteka udostępnia następujące funkcje:

- `int inicjuj()` – zwraca liczbę całkowitą  $k$  ( $1 \leq k \leq 1\,000\,000$ ), określającą, jak wiele projektów ma zawierać ciąg wynikowy. Powinna być użyta dokładnie raz, na samym początku działania programu.
- `int wczytaj()` –  $i$ -te wywołanie zwraca liczbę całkowitą  $p_i$  ( $1 \leq p_i \leq 1\,000\,000\,000$ ) oznaczającą jakość  $i$ -tego projektu (im większa liczba, tym lepszy projekt), albo 0, co oznacza, że nie ma już więcej projektów. Liczba projektów nie jest znana przed wczytaniem danych, jednak możesz założyć, że wszystkich projektów jest przynajmniej  $k$ , a co najwyżej 15 000 000. Funkcja ta powinna być wywoływana do momentu, aż skończą się projekty, **i ani razu więcej**.
- `void wypisz(int jakoscProjektu)` – za pomocą tej funkcji wypisujesz jakości kolejnych projektów, które kanclerz przedstawi królowi. Powinna być ona użyta dokładnie  $k$  razy; w  $i$ -tym wywołaniu należy podać jakość  $i$ -tego w kolejności projektu.  $k$ -te wywołanie tej funkcji zakończy działanie Twojego programu.

## Przykład

Wywołanie	Odpowiedź	Uwagi
<code>k = inicjuj();</code>	3	Od tego momentu $k = 3$ .
<code>wczytaj();</code>	12	
<code>wczytaj();</code>	5	
<code>wczytaj();</code>	8	
<code>wczytaj();</code>	3	
<code>wczytaj();</code>	15	
<code>wczytaj();</code>	8	
<code>wczytaj();</code>	0	0 oznacza koniec wejścia.
<code>wypisz(12);</code>		
<code>wypisz(15);</code>		
<code>wypisz(8);</code>		$k$ -te wywołanie funkcji <code>wypisz</code> kończy program.

## Przykładowa biblioteka

W dziale Pliki znaleźć można przykładową bibliotekę i rozwiązanie ilustrujące sposób ich użycia. Przykładowa biblioteka wczytuje scenariusz testowy ze standardowego wejścia w następującym formacie: Pierwszy wiersz wejścia zawiera jedną dodatnią liczbę całkowitą  $k$ . Kolejne wiersze wejścia zawierają po jednej dodatniej liczbie całkowitej;  $(i + 1)$ -szy wiersz zawiera liczbę  $p_i$ , oznaczającą jakość  $i$ -tego zgłoszonego projektu. Ostatni wiersz wejścia zawiera liczbę 0, oznaczającą koniec listy projektów. Przykładowa biblioteka wypisuje na standardowe wyjście  $k$  wierszy – jakości projektów zgłoszone przez program.

Aby przetestować swój program, skompiluj go komendą podobną do poniższej (zakładamy, że Twoje rozwiązanie znajduje się w pliku `arc.cpp`):

```
g++ -O2 -std=c++11 arc.cpp arczaw.c -o arc
```

Spowoduje to utworzenie pliku wykonywalnego o nazwie `arc`.

## Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Przypominamy, że  $n$  oznacza liczbę projektów.

Podzadanie	Warunki	Liczba punktów
1	$n \leq 30$	10
2	$n \leq 2\,000\,000$	30
3	$n \leq 15\,000\,000$	60

Testy „ocen”:

**1ocen:**  $k = 1, n = 6,$

**2ocen:**  $k = 2, n = 5,$

**3ocen:**  $k = 3, n = 8,$

**4ocen:**  $k = 5, n = 10,$

**5ocen:**  $k = 100\,000, n = 100\,000.$