

## Zadanie C - Operacje tablicowe

Punktów procentowych do uzyskania: **8** (ewentualnie **10**)

Język programowania: C++

autor zadania: Marek Śmieja

Zadanie polega na zaimplementowaniu trzech operacji modyfikujących zadane fragmenty tablicy jednowymiarowej.

### Opis zadania

Aby wyjaśnić dokładne działanie poszczególnych operacji zakładamy, że dana jest  $n$ -elementowa tablica liczb całkowitych o nazwie `data`, przy czym  $n \geq 0$ . Operacje będą wykonywane na fragmencie (ewentualnie na fragmentach) tablicy, to znaczy na podtablicach zadanych za pomocą dwóch liczb całkowitych  $b$  oraz  $e$  spełniających nierówności:  $0 \leq b < n$ ,  $0 \leq e < n$ ,  $b \neq e$ , gdzie  $b$  jest indeksem początku, zaś  $e$  jest indeksem końca fragmentu. Zakładamy ponadto, że tablica `data` reprezentuje „strukturę cykliczną”, to znaczy po ostatnim (o indeksie  $n-1$ ) elemencie następuje element o indeksie 0. Innymi słowy, przejście z ostatniego elementu tablicy na następny element skutkuje przejściem na początek tablicy. W konsekwencji nic nie stoi na przeszkodzie, aby zdefiniować fragment gdzie  $e < b$ . Wówczas fragment dla działań obejmuje elementy o indeksach  $b, b+1, \dots, n-1, 0, 1, \dots, e-1, e$ .

### Specyfikacja możliwych operacji

#### 1. Odwracanie.

W wyniku operacji należy odwrócić kolejność elementów w być może cyklicznym fragmencie od indeksu  $b$  do indeksu  $e$ .

#### 2. Przesunięcie o $k$ pozycji.

Jeśli  $k \geq 0$ , to być może cykliczny fragment od indeksu  $b$  do indeksu  $e$  ulega przesunięciu w prawo o wartość  $k$  miejsc, przy czym przesunięcie mieści się w zadanym fragmencie. Jeśli zaś  $k < 0$  to analogiczny fragment ulega przesunięciu w lewo o  $k$  miejsc.

#### 3. Zamiana fragmentów o zadanej długości.

W wyniku działania operacji należy zamienić ze sobą elementy dwóch być może cyklicznych fragmentów. Pierwszy z nich zadany jest za pomocą indeksów  $b$  oraz  $b+k$ , zaś drugi za pomocą indeksów  $e$  oraz  $e+k$ . Dokładniej mówiąc, każdy element o indeksie  $b+i$  powinien zostać zamieniony z elementem  $e+i$  dla  $i=0,1,\dots,k$ . Dla opisywanej operacji zakładamy,

że fragmenty nie nachodzą na siebie, to znaczy żaden element pierwszego fragmentu nie zawiera się w drugim fragmencie.

### Wejście

- Dane wczytywane są ze standardowego wejścia.
- Pierwsza linia zawiera nieujemną liczbę  $n \leq 10000$  oznaczającą ilość elementów tablicy z danymi.
- W kolejnej linii znajduje się  $n$  liczb całkowitych oddzielonych spacją, stanowiących kolejne elementy tablicy danych.
- Następne linie zawierają kody (pojedynczy znak) oraz liczbowe parametry operacji jakie należy kolejno wykonać oddzielone pojedynczym znakiem spacji. Zakończenie ciągu operacji zadane jest kodem `e`.
- Specyfikacja operacji:
  - Odwracanie oznaczone jest kodem `r` i zawiera dwa parametry  $b$  oraz  $e$ , które specyfikują fragment, który należy odwrócić.
  - Przesunięcie zadane jest kodem `s`. Ponownie parametry  $b$  oraz  $e$  specyfikują fragment, który należy przesunąć. Ponadto, ostatni parametr  $k$  oznacza o ile pozycji należy przesunąć elementy.
  - Zamiana fragmentów jest określona kodem `m`. Kolejne parametry to  $b$  (początek pierwszego fragmentu),  $e$  (początek drugiego fragmentu),  $k$  (długość fragmentu w rozumieniu specyfikacji).
- Operacji do wykonania może być dowolnie wiele.

### Wyjście

Po napotkaniu kodu kończącego sekwencję operacji należy wypisać wynikową tablicę w jednej linii oddzielając elementy pojedynczym znakiem spacji.

### Dodatkowe uwarunkowania

- Pierwsza linia kodu źródłowego MUSI w komentarzu w standardzie języka C++ (dwa znaki ukośnika) zawierać imię i nazwisko autora rozwiązania.
- Jedynym dozwolonym do włączenia plikiem nagłówkowym jest plik `iostream`.
- W rozwiązaniu zabronione jest używanie jakichkolwiek pomocniczych tablic, a jedyna konieczna i dozwolona tablica zawierająca dane musi nazywać się `data`.

- W rozwiązaniu zabronione jest używanie własnych podprogramów.
- Zabronione jest używanie operatora adresu (znak `&`) oraz operatora wyłuskania (znak `*`), ale nie jest zabronione używanie znaków wymienionych operatorów w innych zastosowaniach.
- Jakakolwiek próba obejścia powyższych warunków skutkuje dyskwalifikacją rozwiązania i wyzerowaniem punktacji po upływie terminu oddawania zadania.

### Punktacja

Dla zaliczenia zadania wystarczy przyznanie przez BaCę wartości 80%, co oznacza 8 punktów procentowych z całej puli punktów możliwych do uzyskania na BaCy. Jednakże przyznanie przez BaCę 100% wykonania zadania oznacza dwa bonusowe punkty procentowe z puli punktów BaCy.

### Przykłady wejścia i odpowiadającego wyjścia

Przykład pierwszy:

wejście	wyjście
10	1 8 7 6 5 4 3 2 9 0
0 1 2 3 4 5 6 7 8 9	
r 2 7	
s 8 1 7	
e	

Przykład drugi:

wejście	wyjście
10	5 6 2 3 9 0 1 7 8 4
0 1 2 3 4 5 6 7 8 9	
m 4 9 2	
e	

Przykład trzeci:

wejście	wyjście
10	2 7 8 3 4 5 6 9 0 1
0 1 2 3 4 5 6 7 8 9	
s 7 2 -2	
e	