

ZADANIE 1 (Tablica dwukierunkowa statyczna)

Często w tablicach chcemy mieć możliwość dodawania/usuwania elementów zarówno na początku jak i na końcu. Zadanie polega na zamianie (zwyczajnej) tablicy liczb całkowitych (int) długości $n+3$ w dwukierunkową tablicę zawierającą co najwyżej n elementów (dalej będziemy nazywać taką tablicę jako tablicą dwukierunkową statyczną w skrócie TDS).

W tablicę typu TDS tworzy się ze zwykłej tablicy w ten sposób, że elementy o indeksach 0, 1 oraz 2 nie przechowują danych – służą one jedynie do opisu TDS. Następne elementy mogą zawierać dane dodawane/usuwane przez użytkownika.

0	1	2	n+1	n+2
n	d	s			

Dane będziemy przechowywać na miejscach o indeksach $\{3,4,\dots,n+2\}$, podczas gdy elementy o indeksach 0, 1 oraz 2 opisują strukturę danych.

- Element o indeksie 0 (dalej oznaczany przez n) oznacza maksymalną liczbę elementów, które można przechowywać w *tablicy dwukierunkowej*.
- Element o indeksie 1 (dalej oznaczany przez d) oznacza liczbę elementów aktualnie znajdującą się w *tablicy dwukierunkowej*.
- Element o indeksie 2 (dalej oznaczany przez s) oznacza numer indeksu w wyjściowej tablicy pierwszego elementu w *tablicy dwukierunkowej*.

Dane są przechowywane na miejscach

- od s do $s+d-1$ w przypadku gdy $s+d-1 \leq n+2$,
- do s do $n+2$ oraz od 3 do $s+d-n-1$ w przeciwnym przypadku.

Przyjmujemy dodatkowo, że za elementem o indeksie $n+2$ znajduje się element o indeksie 3 (oznacza to oczywiście, że przed elementem o indeksie 3 jest element o indeksie $n+2$)

Np.:

Tablica

0	1	2	3	4	5	6	7	8	9
7	5	6	-1	0	2	8	4	1	5

Oznacza, że $n=7$, $d=5$, $s=6$. To znaczy TDS mieści maksymalnie 7 elementów, znajduje się w nim 5 elementów przy czym pierwszy element jest element z indeksem 6 (w tym przypadku jest to 8).

Ponadto, zgodnie z opisem, po elemencie z indeksem 9 jest element z indeksem 3.

Tak więc przedstawiony powyżej TDS zawiera elementy 8, 4, 1, 5, -1.

Natomiast TDS:

0	1	2	3	4	5	6	7	8	9
7	3	5	-1	0	2	8	4	1	5

Zawiera 3 elementy począwszy od tego znajdującego się w tablicy pod indeksem 5 (w tym przypadku 2, 8, 4.

Zadanie polega na utworzeniu funkcji obsługujących TDS:

1. `void initialize(int *t, int n)`
z danej tablicy `t` długości `n` tworzy pusty TDS mogący zawierać co najwyżej `n-3` elementów
2. `bool isfull(int *t)`
sprawdza czy TDS `t` zawiera wolne miejsca (to znaczy czy można jeszcze dodać do niego elementy)
3. `bool isempty(int *t)`
sprawdza czy TDS `t` jest pusty (to znaczy zwraca `true` wtedy i tylko wtedy, gdy `t` nie zawiera elementów)
4. `bool addLast(int *t, int k)`
dodaje liczbę `k` za ostatnim elementem w TDSie `t`.
Zwraca `true` jeśli tablica nie była pełna, zwraca `false` jeśli tablica w momencie wywołania funkcji była pełna (w takim przypadku element `k` nie jest dodawany).
5. `bool addFirst(int *t, int k)`
dodaje liczbę `k` przed pierwszym elementem w TDSie `t`.
Zwraca `true` jeśli tablica nie była pełna, zwraca `false` jeśli tablica w momencie wywołania funkcji była pełna (w takim przypadku element `k` nie jest dodawany).
6. `bool delFirst(int *t, int &el)`
ustawia na `el` wartość pierwszego elementu TDS-u a następnie usuwa ten element ze struktury.
Zwraca `true` jeśli tablica była niepusta, zwraca `false` jeśli tablica w momencie wywołania funkcji była pusta – w takim przypadku wartość `el` może być dowolna, tablica powinna pozostać pusta.
7. `bool delLast(int *t, int &el)`
ustawia na `el` wartość ostatniego elementu TDS-u a następnie usuwa ten element ze struktury.
Zwraca `true` jeśli tablica była niepusta, zwraca `false` jeśli tablica w momencie wywołania funkcji była pusta – w takim przypadku wartość `el` może być dowolna, tablica powinna pozostać pusta.
8. `int iterator(int *t)`
zwraca numer w tablicy `t` **poprzedzający** pierwszy element TDS-u `t`.
9. `bool next(int *t, int &it)`
Jeśli element `it` nie jest ostatnim elementem w TDS-ie `t` to zmienia wartość `it` tak aby wskazywała kolejny element oraz zwraca `true`. W przeciwnym razie zwraca `false`, wartość `it` może być dowolna.
10. `void out(int *t)`
Wypisuje wszystkie elementy w TDS-ie `t` (od pierwszego do ostatniego). Elementy powinny być oddzielone pojedynczym odstępem, po ostatnim powinien znajdować się znak nowej linii.
11. `void outDBG(int *t)`
wypisuje wszystkie elementy w tablicy `t` – rozumianej jako zwykła tablica – NIE TDS.
12. `int min(int *t)`
zwraca najmniejszy element w TDS-ie `t`. Działanie funkcji dla pustego TDSa nie jest określone.
13. `int max(int *t)`
zwraca największy element w TDS-ie `t`. Działanie funkcji dla pustego TDSa nie jest określone.
14. `float avg(int *t)`
zwraca średnią arytmetyczną elementów w TDS-ie `t`. Działanie funkcji dla pustego TDSa nie jest określone.

Uwagi:

Plik powinien nazwać się *zadanie1_nazwisko.cpp*

Plik ma zawierać:

- wskazane powyżej funkcje,
- niezbędne biblioteki (include),
- linię z podpisem zapisaną bez polskich liter w następujący sposób :
`#define AUTOR „Sławomir Kowalski”`

Plik ma NIE zawierać funkcji main!!

Rozwiązania należy przesyłać pod adres ppasteczka@up.krakow.pl do dnia

10.04 godziny 23:59:59.

Jeśli w terminie 2 dni nie przyjdzie potwierdzenie otrzymania proszę o przesłanie ponownie.

Wszelkie pytania proszę kierować pod adres: ppasteczka@up.krakow.pl

Odpowiedzi są umieszczane pod adresem <http://ppasteczka.up.krakow.pl/dydaktyka/FAQ.html>