Halinka na schodach

Wskazówki do rozwiązania zadania

Numery schodów wczytujemy do tablicy a na pozycje o numerach $i = 1, 2, \ldots, n$. Podczas wczytywania zliczamy, ile razy na wejściu występuje liczba 1 – to będzie ilość zestawów schodów odwiedzonych przez Halinkę (każdy zestaw zaczyna się od numeru 1).

Następnie ponownie przeglądamy liczby a_i, $i = 2, 3, \ldots, n$ – jeśli a_i jest równe 1, to znaczy, że właśnie zaczynamy nowy zestaw schodów, a wartość a_{i-1} to ostatni schodek w poprzednim zestawie, czyli zarazem wielkość tamtego zestawu, którą wypisujemy na ekranie. W ten sposób wypiszemy rozmiary wszystkich zestawów z wyjątkiem ostatniego, którego rozmiar jest równy po prostu a_n. Tę wartość wypisujemy na koniec.

Kod źródłowy rozwiązania w języku C++

```
#include <iostream>
using namespace std;

int main()
{
  int n; cin >> n;
  int a[10001];
  int il = 0;
  for(int i = 1; i <= n; i++)
  {
    cin >> a[i];
    if(a[i] == 1) il++;
  }
  cout << il << endl;
  for(int i = 2; i <= n; i++)
    if(a[i] == 1)
      cout << a[i - 1] << ' ';
  cout << a[n] << endl;;
  return 0;
}
```

Uwagi

Tablicę $a[\,]$ deklarujemy o jeden element większą od największego możliwego n. Dzieje się tak dlatego, że numeracja elementów tablicy zaczyna się od indeksu 0, natomiast numeracja ciągu liczb w zadaniu (a_i) zaczyna sie od indeksu 1. (Dokładniejsza dyskusja na ten temat przedstawiona jest w scenariuszu zajęć z zadaniem *Plan lekcji*.)

Istnieje alternatywny sposób znalezienia długości ostatniej klatki schodowej – przy użyciu tak zwanego *wartownika*. Zwróćmy uwagę, że najwyższy stopień klatki rozpoznajemy po tym, że zaraz za nim następuje stopień o numerze 1. Tylko ostatnia klatka nie spełnia tego warunku, bo po niej nie ma już stopni. Można uciec się do następującego tricku: dopiszemy *po* ostatniej klatce fikcyjny stopień o numerze 1 i teraz zakończenie każdej klatki będzie wyglądało tak samo. Taka dodatkowa wartość dodana do istniejącej struktury nosi nazwę *wartownika*. Wymaga to jednak zadeklarowania tablicy $a[\,]$ większej jeszcze o jedną komórkę – tak, aby było miejsce na wartownika. Kod programu-rozwiązania należy nieco zmodyfikować:

```
. . .
int a[10002];
. . .
a[n + 1] = 1; // dodajemy wartownika
for(int i = 1; i <= n; i++)
  if(a[i + 1] == 1) // jeśli następny stopień ma numer 1,
                    // jesteśmy na szczycie klatki
    cout << a[i] << ' ';
cout << endl;
. . .
```