

Zajęcia 6

Temat: Tablice 2

Czas trwania: 2x45 min

Cel zajęć:

projektuje i programuje proste problemy z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, tablice, testuje poprawność programów dla różnych danych, posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

Efekty:

- umie uruchomić potrzebne oprogramowanie,
- umie napisać program z wykorzystaniem instrukcji warunkowej i iteracyjnej i tablic w C++,

Formy i metody pracy: praca samodzielna, omówienie, wykład

Zadania do wykonania na zajęciach	Treści programowe
1. Pociąg	M.2, P.2.13, A.3.2
2. Bitib, Bajtjab i Palindrom	M.2, P.2.13, A.3.2
3. Odciski palca	M.2, P.2.13, A.3.2

Materiały do zajęć:

<https://www.main2.edu.pl/main2/courses/show/6/17/>

Zadania do wykonania w domu:

Żarówki (zliczanie):

<https://szkopul.edu.pl/c/plo155/p/zar/25364/>

Przyciski (VI OIG – zliczanie i maksimum):

<https://szkopul.edu.pl/problemset/problem/aPqgk8oaUsM4nB6FLjxehRPe/site/?key=statement>

Zapałki (V OIG, sumy prefiksowe od lewej i prawej):

https://szkopul.edu.pl/problemset/problem/ZLG7FB_afACLMh8-zsupw5zV/site/?key=statement

ZADANIA I ROZWIĄZANIA

Zadanie 1. Pociąg

Dostępna pamięć: 64MB

„Stoi na stacji lokomotywa,
Ciężka, ogromna i pot z niej splywa -
Tłusta oliwa.
[...]
A tych wagonów jest ze czterdzieści,
Sam nie wiem, co się w nich jeszcze mieści.
Lecz choćby przyszło tysiąc atletów
I każdy zjadłby tysiąc kotletów,
I każdy nie wiem jak się natęzał,
To nie udźwigną - taki to ciężar!”

Julian Tuwim, Fragment wiersza 'Lokomotywa'

Jaś pracuje na rampie kolejowej i przeładowuje wagony. Wiedząc, jak ciężkie mogą być wagony, postanowił zakupić specjalny dźwig. Może on podnieść dowolną liczbę wagonów jednocześnie, ważne, żeby stały one obok siebie. Niestety, nawet taki super-sprzęt posiada maksymalny udźwig. Jak duży ciężar musi podnieść dźwig Jasia?

Wejście

W pierwszym wierszu wejścia znajduje się liczba wagonów n pewnego pociągu ($2 \leq n \leq 10^6$). W drugiej linii znajduje się n liczb całkowitych w_i – ciężar każdego z wagonów ($1 \leq w_i \leq 10_6$). W trzeciej linii znajduje się jedna liczba całkowita k – liczba zestawów wagonów ($1 \leq k \leq 10^6$). W kolejnych k liniach znajdują się informacje o zestawach wagonów, które należy podnieść dźwigiem, odpowiednio numer pierwszego w_p i ostatniego wagonu w_k do podniesienia (należy jednocześnie podnieść wszystkie wagony od w_p do w_k włącznie).

Wyjście

Twój program powinien w k liniach ciężar każdego z zestawów wagonów.

Przykład

Wejście	Wyjście
5	9
3 5 4 5 4	13
2	
2 3	
3 5	

Rozwiązanie

Najprostszym i najbardziej intuicyjnym rozwiązaniem wydaje się zliczanie ciężarów wagonów od pierwszego w_p do ostatniego wagonu w_k (dla uproszczenia zadania wagony numerujemy od 1).

```
wczytaj n
dla i=1 do n wykonaj
    wczytaj a[i]
wczytaj k
```

```

dla i=1 do k wykonaj
    wczytaj wp, wk
    suma ← 0
    dla j=wp do wk wykonaj
        suma ← suma + a[j]
    wypisz suma

```

Zadanie dla uczniów: Uzasadnij, że liczba operacji rośnie wraz z kwadratem danych wejściowych.

Rozwiązanie szybkie: Wykorzystajmy fakt, że łatwo jest już w trakcie wczytania policzyć, ile waży wszystkie wagony od pierwszego do aktualnie wczytanego (wystarczy do dotychczasowej sumy dodać ciężar aktualnego wagonu). Przykład:

nr wagonu	0	1	2	3	4	5	6	7	8	9	10
ciężar wagonu:	0	1	1	2	1	1	2	1	1	1	2
suma ciężarów wszystkich wagonów od pierwszego do aktualnego:	0	1	2	4	5	6	8	9	10	11	13

Zauważ, że aby obliczyć ciężar wagonów od w_p do w_k wystarczy odjąć od sumy ciężarów wagonów do w_k sumę wagonów znajdujących się przed w_p .

```

wczytaj n
dla i=1 do n wykonaj
    wczytaj a[i]
    suma[i] ← suma[i-1] + a[i]
wczytaj k
dla i=1 do k wykonaj
    wczytaj wp, wk
    wypisz suma[wk] - suma[wp - 1]

```

Zadanie dla uczniów: Ile operacji dodawania i odejmowania w zależności od n i od k wykona nasz algorytm?

Zadanie 2. Bitib, Bajtjab i Palindrom

Dostępna pamięć: 64MB.

Bitib i Bajtjab lubią takie ciągi liczb, które obaj mogą czytać jednocześnie - Bitib od pierwszej do ostatniej, zaś Bajtjab od ostatniej do pierwszej, a mimo to czytają to samo. Czy liczby na kartce, którą właśnie znaleźli, można tak poprzestawiać, aby otrzymać ulubiony przez chłopców sposób ich ustawienia?

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna n nie większa niż milion. W następnej linii znajduje się n liczb naturalnych nie większych niż milion.

Wyjście

Jedyny wiersz wyjścia zawiera słowo „TAK” – jeśli z elementów ciągu można zbudować palindrom, lub „NIE” – jeśli nie jest to możliwe.

Przykład

Wejście 5 15 15 81 18 81 Wyjście TAK	Wejście 6 15 15 15 15 18 81 Wyjście NIE
Wyjaśnienie: Przykładowy możliwy do uzyskania palindrom: 81 15 18 15 81	

Rozwiązanie

Spostrzeżenie: Aby było możliwe czytanie liczb od początku i od końca w tym samym momencie, dana liczba musi wystąpić dokładnie dwa razy. Zatem jeżeli liczba występuje więcej niż dwa razy, musimy zaobserwować parzystą liczbę wystąpień. Jedyny wyjątek stanowi liczba o nieparzystej liczbie wystąpień w przypadku nieparzystej n .

Zatem musimy policzyć, ile razy wystąpiła każda z liczb. Zapamiętajmy też największą liczbę, jaka znalazła się na kartce (wykorzystujemy w drugiej pętli):

```
wczytaj n
największa ← 0
dla i=1 do n wykonaj
    wczytaj x
    ile_razy_wystąpiła[x] ← ile_razy_wystąpiła[x] + 1
    jeżeli x > największa
        największa ← x
ile_nieparzystych ← 0
dla i=1 do największa wykonaj
    jeżeli ile_razy_wystąpiła[i] mod 2 = 0
        ile_nieparzystych ← ile_nieparzystych + 1
jeżeli ile_nieparzystych > 1
    wypisz NIE
w przeciwnym wypadku
    wypisz TAK
```

Zadanie dla uczniów: Jaka jest złożoność obliczeniowa programu? Jakie warunki muszą być spełnione, by takie rozwiązanie było możliwe?

Zadanie 3. Odciski palca

Dostępna pamięć: 32MB

Pan Integer pracuje nad programem do odczytywania odcisków palców. Najwięcej uwagi poświęca modułowi odpowiedzialnemu za sprawdzanie, czy pewien kwadratowy

czteroelementowy fragment odnalezionego obrazu (zdjęty z przedmiotu odcisk) może być jakąś częścią dużego obrazu (pełnego wzoru). Zastanawia się przy tym, ile razy fragment pojawił się w jakiegokolwiek formie na dużym obrazie. Pomóż mu to obliczyć!

Wejście

W pierwszej i drugiej linii wejścia znajdują się po dwie dwucyfrowe liczby pierwsze – odnaleziony fragment obrazu. W drugiej linii wejścia znajdują się dwie liczby całkowite h oraz w ($1 \leq h, w \leq 1000$), odpowiednio wysokość i szerokość dużego obrazu.

W kolejnych h liniach znajduje się po w rozdzielonych spacją dwucyfrowych liczb pierwszych.

Wyjście

Liczba wystąpień dowolnej permutacji fragmentu obrazu we wzorze.

Przykład

Wejście 11 13 17 19 2 3 23 11 13 29 17 19 Wyjście 1	Wejście 11 13 17 19 3 4 23 11 13 17 29 17 19 11 11 17 17 13 Wyjście 3	Wejście 11 13 17 19 1 4 11 13 17 19 Wyjście 0
--	---	---

Rozwiązanie

Zadanie dla uczniów: Uzasadnij, że iloczyn czterech dowolnych liczb pierwszych (niekoniecznie różnych) będzie unikatowy (to znaczy, że nie można znaleźć innej czwórki liczb pierwszych, której iloczyn miał taką samą wartość).

Korzystając z powyższego spostrzeżenia wystarczy sprawdzać kolejne kwadratowe obszary, czy ich iloczyn jest równy iloczynowi wzorcowej czwórki.

```
wczytaj p1, p2
wczytaj p3, p4
palec ← p1 · p2 · p3 · p4
wczytaj h, w
dla y=1 do h wykonaj
    dla x=1 do w wykonaj
        wczytaj a[x][y]
wynik ← 0
dla y=1 do h-1 wykonaj
    dla x=1 do w-1 wykonaj
        jeżeli palec = a[x][y] · a[x][y+1] · a[x+1][y] · a[x+1][y+1]
            wynik ← wynik + 1
wypisz wynik
```