

Zajęcia 5

Temat: Tablice 1

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, iteracyjnej i tablic w C++,
- umie wyczytywać dane do tablicy i wykonywać proste operacje,

Formy i metody pracy: praca samodzielna,

Zadania do wykonania na zajęciach	Treści programowe
1. Temperatury	M.2, P.2.13, A.3.2
2. Najlepsze sumy	M.2, P.2.13, A.3.2
3. Bankiet	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:

Średnia:

<https://szkopul.edu.pl/problemset/problem/YYAbpdvNnjQKBQQEdhKHKR7W/site/?key=statement>

Szuflady:

<https://szkopul.edu.pl/problemset/problem/ERbofDxdUsKn2g8Tnwc25t6R/site/?key=statement>

Krażki:

<https://szkopul.edu.pl/problemset/problem/fYXVXOreVxiXTRoHZJXyXF2l/site/?key=statement>

ZADANIA I ROZWIĄZANIA

Zadanie 1. Temperatury

Dostępna pamięć: 256MB

Bajtek na lekcji przyrody uczy się określać różne zjawiska pogodowe. Jednym z zadań, które otrzymał, jest codzienne notowanie temperatury. Bajtek rzetelnie zapisywał liczby na karteczce każdego dnia. Teraz przygląda się uzyskanym wynikom i zastanawia, ile razy i w które dni uzyskał określoną temperaturę. Pomożesz mu?

Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita n ($1 \leq n \leq 10^3$), oznaczająca liczbę dni pomiarowych. W drugiej linii wejścia znajduje się n liczb całkowitych - zanotowanych w kolejnych dniach temperatur t_i ($-50 \leq t_i \leq 100$). W trzeciej linii znajduje się jedna liczba x ($-50 \leq x \leq 100$) - szukana temperatura.

Wyjście

Na wyjściu w jednej linii powinna znaleźć się liczba dni k , w które została zmierzona temperatura x , oraz k liczb oznaczających numery dni, w które zmierzono temperaturę x . Liczby należy podać w kolejności rosnącej i oddzielić pojedynczymi spacjami.

Przykład

Wejście 5 -2 0 1 -2 3 -2	Wyjście 2 1 4
-----------------------------------	------------------

Rozwiązanie

Rozwiązanie zadania wymaga omówienia podstawowych informacji związanych z przetwarzaniem i przeszukiwaniem tablic jednowymiarowych.

Wymuszone zostało dwukrotne przeszukanie tablicy: zliczenie wartości oraz wypisanie indeksów.

```
wczytaj n
dla i=1 do n wykonaj
    wczytaj a[i]
ile ← 0
wczytaj x
dla i=1 do n wykonaj
    jeżeli a[i] = x
        ile ← ile + 1
wypisz ile
dla i=1 do n wykonaj
    jeżeli a[i] = x
        wypisz i
```

Zadanie 2. Najlepsze sumy

Dostępna pamięć: 32MB. Na podstawie zadania cke.gov.pl

Najlepszą sumą ciągu liczb a_1, a_2, \dots, a_n nazywamy największą wartość wśród sum złożonych z sąsiednich elementów tego ciągu. Na przykład dla ciągu: 1, 2, -5, 7 mamy następujące sumy:

1
1+2 = 3
1+2+(-5) = -2
1+2+(-5)+7 = 5
2
2+(-5) = -3
2+(-5)+7 = 4
-5
-5+7 = 2
7

Zatem najlepszą sumą jest 7 (zwróć uwagę, że jeden element też uznajemy za sumę).

Zaproponuj algorytm wyznaczania najlepszej sumy dla dowolnego ciągu liczb całkowitych. Na jego podstawie napisz program do obliczenia najlepszych sum ciągów liczb.

Wejście

W jednej linii znajduje się ciąg liczb całkowitych zakończony liczbą 0. Liczb jest nie więcej niż 500000, ich wartość mieści się w przedziale $[-1000, 1000]$.

Wyjście

Największa suma złożona z kolejnych elementów ciągu.

Przykład

Wejście	Wyjście
1	7
2	
-5	
7	
0	

Rozwiązanie

Zauważmy, że w naszym wypadku wynik nie będzie nigdy gorszy niż 0 kończące ciąg. Najprostszym rozwiązaniem wydaje się policzenie wszystkich możliwych sum i wybranie największej z nich. Niech k będzie liczbą elementów.

```
k ← 1
wykonaj
    k ← k + 1
    wczytaj a[k]
dopóki a[k] ≠ 0
maksimum ← 0
```

```

dla i=1 do k wykonaj
    suma ← 0
    dla j=i do k wykonaj
        suma ← suma + a[j]
        jeżeli suma > maksimum
            maksimum ← suma
wypisz maksimum

```

Zadanie dla uczniów: Ile operacji sumowania wykona nasz program?

W pierwszym przebiegu zewnętrznej pętli wykona k dodawań, w drugim $k - 1$, w trzecim $k - 2$, ... w k -tym – jedno. Łącznie więc zostanie wykonanych $k + k - 1 + k - 2 + \dots + 1 = (k \cdot (k+1)) \div 2$ operacji. Widać więc, że liczba operacji rośnie wraz z kwadratem rozmiaru danych wejściowych.

Rozwiązanie optymalne zakłada sumowanie kolejnych elementów dopóki się to „opłaca”, to znaczy dopóki suma częściowa jest większa od 0. W przeciwnym wypadku opłaca się pozbyć zmniejszającej końcowy wynik wartości i liczyć od nowa.

```

k ← 1
wykonaj
    k ← k + 1
    wczytaj a[k]
dopóki a[k] ≠ 0
maksimum ← 0
suma ← 0
dla i=1 do k wykonaj
    suma ← suma + a[i]
    jeżeli suma > maksimum
        maksimum ← k
    jeżeli suma < 0
        suma ← 0
wypisz maksimum

```

Zadanie dla uczniów: Ile operacji sumowania wykona nasz program?

Zadanie dla uczniów: Dokonaj analizy liczby wykonywanych operacji w zależności od rozmiaru danych wejściowych w dotychczas wykonanych zadaniach: Dzielniki, Ile podzielnych przez 3 i przez 5, Podstawowa arytmetyka oraz Najlepsze sumy.

Dla wnikliwych uczniów: Czy w ogóle była nam potrzebna tablica?

Zadanie 3. Bankiet

Dostępna pamięć: 32MB. I OIG (dostępne w serwisie szkopuł.edu.pl)

W restauracji Utalentowany Miś zaplanowano bankiet dla finalistów OIG. Goście zasiądą przy okrągłych stołach w ściśle określony sposób. Kierownik sali otrzymał listę gości wraz z informacją, kto ma siedzieć z lewej strony każdego z nich. Ile stołów musi przygotować na bankiet?

Zadanie

Opracuj program, który:

- wczyta ze standardowego wejścia informacje o rozmieszczeniu gości,
- obliczy ile stołów trzeba przygotować,
- wypisze wynik na standardowe wyjście.

Wejście

W pierwszym wierszu zapisano liczbę gości N ($1 \leq N \leq 30\,000$). Goście są ponumerowani kolejnymi liczbami naturalnymi od 1 do N . W drugim wierszu zapisano numer gościa siedzącego po lewej stronie pierwszego gościa. W trzecim wierszu zapisano numer gościa siedzącego po lewej stronie drugiego gościa itd. W i -tym wierszu zapisano numer gościa siedzącego po lewej stronie $(i-1)$ -tego gościa. W $N+1$ -szym wierszu zapisano numer gościa siedzącego po lewej stronie N -tego gościa.

Wyjście

W pierwszym wierszu wypisz liczbę stolików potrzebnych do usadzenia wszystkich gości.

Przykład

Wejście	Wyjście
12	4
4	
10	
7	
3	
2	
6	
1	
5	
11	
8	
12	
9	

Rozwiązanie

Otrzymaliśmy już gotową listę (rzeczywiste ustawienie), więc możemy założyć, że każde dziecko wskazało kogoś innego. Wystarczy więc przeprowadzić symulację kolejnych dzieci aż natrafimy na już usadzone (będziemy oznaczać je liczbą 0). Liczba symulacji to szukana liczba stołów. Algorytm wygląda więc następująco:

```
wczytaj n
dla i=1 do n wykonaj
    wczytaj a[i]
ile ← 0
dla i=1 do n wykonaj
    jeżeli a[i] ≠ 0
        ile ← ile + 1
        k ← a[i]
        dopóki k ≠ 0
```

```
j ← k  
k ← a[j]  
a[j] ← 0
```

wypisz ile

Zadanie dla uczniów: Ile operacji sumowania wykona nasz program mimo „pętli w pętli”?
Odpowiedź uzasadnij.