

## Zajęcia 18

**Temat:** Struktury danych 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, rekurencję, pisze własne funkcje rekurencyjne, zbiory i operacje na zbiorach, struktury danych, biblioteka STL, 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 funkcji rekurencyjnej, równaniami rekurencyjnymi,
- zna struktury danych: stos, kolejkę, vector, pair, kolejka priorytetowa

**Formy i metody pracy:** praca samodzielna, wykład, omówienie

Zadania do wykonania na zajęciach	Treści programowe
1. Mnoż, odejmuj, dodawaj (mod)	M.3, P.2.16, P.2.18, A.4.1, A.4.2
2. Wodzirej (wod)	M.3, P.2.16, P.2.18, A.4.2, A.4.5
3. Gorące pudełko (gpuzad)	M.3, P.2.16, P.2.20, A.4.5, A.4.9

**Materiały do zajęć:**

<https://en.cppreference.com/w/>

**Zadania do wykonania w domu:**

**Plakatowanie (OI)**

<https://szkopul.edu.pl/problemset/problem/au-E9FH96-3U9rCKhcNsD5n9/site>

## ZADANIA I ROZWIĄZANIA:

### Zadanie 1. Mnóż, odejmuj, dodawaj

Limit pamięci: 32MB

Czy dodawanie, odejmowanie i mnożenie może sprawiać kłopoty? Oczywiście że nie! Dlatego mam dla ciebie proste zadanie: dla podanego w specjalnym zapisie wyrażenia arytmetycznego oblicz jego wartość! Specjalność tego zapisu polega na tym, że najpierw zawsze pojawiają się argumenty, a dopiero później działania M, O lub D (mnożenie, odejmowanie lub dodawanie). W wyrażeniu tym jednak nie musisz się martwić kolejnością działań arytmetycznych – po prostu wykonuj je zgodnie z wejściem!

#### Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba  $N$  ( $1 \leq N \leq 10$ ) – liczba wyrażeń do rozpatrzenia. W kolejnych  $N$  liniach znajdują się krótkie wyrażenia arytmetyczne. Długość żadnego z nich nie przekracza 1000 znaków. Wyrażenia składają się wyłącznie z wartości  $x_i$  ( $0 \leq x_i \leq 10^{15}$ ) oraz działań opisanych odpowiednimi literami.

#### Wyjście

W oddzielnych liniach należy wypisać  $N$  liczb całkowitych – wyniki obliczeń. Możesz założyć, że końcowy wynik nigdy nie przekroczy  $|10^{15}|$ .

#### Przykład

Wejście	Wyjście
5	9
4 5 D	-10
10 20 O	11
2 3 M 5 D	17
2 3 5 M D	9
5 5 M 20 O 2 2 D D	

#### Wyjaśnienie:

Kolejne wyrażenia to odpowiednio:  $4+5$ ,  $10-20$ ,  $2*3+5$ ,  $2+3*5$ ,  $5*5-20+2+2$

#### Rozwiązania

W zadaniu mamy do czynienia z zapisem wyrażeń arytmetycznych w odwrotnej notacji polskiej. Aby rozwiązać to zadanie wykorzystamy stos (można wykorzystać `stack <long long>` z biblioteki STL). Wyznaczanie wartości wyrażenia odbywa się według poniższego algorytmu (dla jednej linii tekstu):

```
stos liczb całkowitych S
dla wszystkich elementów w linii
    pobierz kolejny element z wejścia
    jeżeli element jest liczbą
        dodaj liczbę na stos
    jeżeli element jest operacją arytmetyczną
        zdejmij ze stosu ostatnia liczbę x1
        zdejmij ze stosu ostatnia liczbę x2
```

```

jeżeli element = 'M'
    dodaj na stos x1*x2
jeżeli element = 'D'
    dodaj na stos x1+x2
jeżeli element = 'O'
    dodaj na stos x2-x1
zdejmij ze stosu ostatnia liczbę x
wypisz x

```

Zadanie dla uczniów: Jak można przy wczytywaniu liczb wykorzystać schemat Hornera?

## Zadanie 2. Wodzirej

Dostępna pamięć: 32 MB

Janek urządził imprezę. Tym razem postanowił wybrać jedną osobę, która będzie decydowała o przebiegu zabaw – wodzireja. Chciałby, aby osoba ta była lubiana przez jak największą grupę uczestników spotkania. Pomóż wskazać mu tę osobę oraz grono osób, które ją lubią.

### Wejście

Pierwszy wiersz danych zawiera dwie liczby całkowite  $n$  i  $m$ , odpowiednio liczbę osób biorących udział w przyjęciu oraz liczbę wzajemnych sympatii ( $2 \leq n \leq 10\,000$ ,  $2 \leq m \leq 1\,000\,000$ ). W kolejnych  $m$  liniach znajdują się po dwie liczby całkowite  $u_i$  i  $v_i$  ( $1 \leq u_i, v_i \leq 10\,000$ ) – informacje o parach osób, które się lubią wzajemnie.

### Wyjście

Program powinien wypisać w pierwszym wierszu jedną liczbę – najbardziej lubianą osobę na przyjęciu (wodzireja). Jeśli jest więcej takich osób – tę o najmniejszej liczbie porządkowej.

W drugiej linii program powinien wypisać w kolejności rosnącej wszystkie osoby, które lubią wodzireja.

### Przykład

Wejście	Wyjście
5 4	2
1 3	3 4 5
2 3	
4 2	
5 2	

## Rozwiązanie

Ponieważ nie możemy z góry założyć, z iloma uczestnikami lubi się każdy z imprezowiczów, wykorzystajmy do zapamiętywania znajomych tablicy dynamicznej (można wykorzystać `vector<int>` z biblioteki STL). Następnie odnajdziemy pierwszy najdłuższy ciąg, posortujemy go i wypiszemy. Poniżej kod zadania w C++ wraz komentarzami:

```
stos liczb całkowitych S
```

```
//lista znajomych
```

```

vector <int> V[10001];
cin >> n >> m;
//dla wszystkich par sympatii
for (int i=0; i<m; i++) {
    //wczytaj numery pary lubiących się wzajemnie znajomych
    cin >> u >> v;
    //dodaj do listy znajomych u znajomego v
    V[u].push_back(v);
    //dodaj do listy znajomych v znajomego u
    V[v].push_back(u);
}
//zakładamy, że nikt nie jest najbardziej lubiany
maximum = 0;
//dla wszystkich imprezowiczów
for (int i=1; i<=n; i++)
    //jeżeli jakiś imprezowicz ma więcej lubiących go znajomych
    //niż dotychczasowy najlepszy imprezowicz, zapamiętaj jego numer
    if (V[maximum].size() < V[i].size()) maximum = i;
//wypisz numer najbardziej lubianego imprezowicza
cout << maximum << endl;
//posortuj jego znajomy numerami rosnąco
sort ( V[maximum].begin(), V[maximum].end());
//wypisz wszystkich znajomych naszego ulubieńca
for (int i=0; i< V[maximum].size(); i++)
    cout << V[maximum][i] << " ";

```

Uwaga: Warto wrócić do treści tego zadania na zajęciach nr 21 (grafy) i wykazać, że w zadaniu wodzirej zaimplementowaliśmy już graf.

### Zadanie 3. Gorące pudełko

Dostępna pamięć: 256MB

Czarująca pani Char, przedszkolanka w przedszkolu Słoneczny Integer, lubi urządzać dla swoich podopiecznych różne zabawy. Jedną z nowych zabaw, które pani Char zaproponowała dzieciom, jest zabawa w gorące pudełko. Polega ona na tym, że pani Char wręcza pierwszemu dziecku kolorowe pudełko, ono podaje je drugiemu, drugie - trzeciemu, itd.. Ostatnie dziecko podaje pudełko pierwszemu.

W czasie zabawy gra muzyka. Gdy melodia ucichnie, dziecko, które trzyma pudełko, dostaje pyszne ciasteczko i kończy zabawę. Pani Char przerywa muzykę w różnych nieoczekiwanych dla dzieci momentach. Pudełko porusza się raz w prawo, raz w lewo. Zabawa trwa tak długo, aż zostanie tylko jedno dziecko - zwycięzca gry.

Każdy z maluchów chciałby oczywiście wygrać zabawę. Mały Bitek zauważył w pewnym momencie, że pani Char zawsze używa takiej samej sekwencji ruchów. Zauważył też, że może wybrać takie miejsce wśród dzieci, które zapewni mu wygraną. Zastanawia się teraz, które to miejsce. Pomożesz mu?

Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita  $n$  ( $2 \leq n \leq 10^6$ ), oznaczająca liczbę dzieci w przedszkolu.

W kolejnych  $n-1$  liniach wejścia znajduje się sekwencja ruchów wykonywanych przez panią Char, przy czym  $i$ -ty ruch opisany jest przez liczbę całkowitą  $x_i$  ( $1 \leq x_i \leq 100$ ) oznaczającą liczbę przekazania przez dzieci pudełka, po których ucichnie muzyka. Ruchy wykonywane są na przemian raz w prawo, raz w lewo, przy czym pierwszy ruch wykonywany jest zawsze w prawo. Dziecko, które odpadło, oddaje pudełko temu dziecku, od którego je otrzymało.

### Wyjście

Na wyjściu powinna znaleźć się jedna liczba całkowita oznaczająca liczbony od lewej numer miejsca, które powinien zająć mały Bitek, żeby wygrać zabawę.

### Przykład

Wejście	Wyjście
5	3
3	
2	
5	
1	

Objaśnienie przykładu: Pogrubieniem oznaczmy położenie pudełka. Na początku otrzymuje gorące pudełko zawsze pierwsze dziecko: [1,2,3,4,5]. Muzyka cichnie po trzech podaniach, więc odpada dziecko czwarte: [1,2,3,5]. Teraz pudełko przechodzi o dwa w lewo, odpada dziecko pierwsze: [2,3,5]. Dalej - cyklicznie - pięć razy w prawo - odpada piąte: [2,3]. I raz w lewo - zostało tylko dziecko trzecie: [3].

### Ocenianie

Podzadanie	Ograniczenia	Punkty
1	$n \leq 10^3$	30
2	brak dodatkowych założeń	70

### Rozwiązanie

Zauważmy, że często będziemy usuwać elementy ze zbioru. W zadaniu tym możemy więc skorzystać strukturę reprezentującą zbiór (np. set z STL). Ciekawszym rozwiązaniem będzie jednak wykorzystanie listy cyklicznej (użyjemy `list<int>` z biblioteki STL pamiętając, by zawsze ostrożnie przechodzić pomiędzy pierwszym i ostatnim elementem listy). Kod programu stanowi symulację przekazywania pudełka pomiędzy dziećmi.

```
list<int> lista;
list<int>::iterator p, q;
int n, x;
//dodaj n dzieci do listy
cin >> n;
for (int i=1; i<=n; i++) lista.push_back(i);
//zapamiętaj początek listy
p=lista.begin();
//dla n-1 dzieci
for (int i=0; i<n-1; i++){
    //wczytaj liczbę dzieci, które przekażą sobie pudełko
    cin >> x;
    //idź w prawo o jeden ruch za mało, iterator zwiększaj
```

```

if (i%2==0) {
    for (int j=1; j<x; j++){
        p++;
        //jeżeli doszedłeś do końca listy dzieci, wróć na początek
        if (p==lista.end()) p=lista.begin();
    }
    //dziecko q do usunięcia z kolejki jest za dzieckiem p
    q=p;
    q++;
    //ostrożnie z końcem listy
    if (q==lista.end()) q=lista.begin();
    //teraz możesz usunąć dziecko z kolejki
    lista.erase(q);
}
//podobnie idziesz w lewo, iterator zmniejszasz
else {
    for (int j=1; j<x; j++){
        //również ostrożnie z przejściem przez początek-koniec listy
        if (p==lista.begin()) p=lista.end();
        p--;
    }
    //usuń z kolejki poprzednie dziecko
    q=p;
    if (q==lista.begin()) q=lista.end();
    q--;
    lista.erase(q);
}
//w kolejce zostało jedno dziecko, wypisz go
cout << lista.front();

```