

**Zadanie 1 (10 pkt. na pracowni, później 5 pkt.)**

Napisz program, który policzy długość linii łamanej zamkniętej, której kolejne wierzchołki znajdują się w punktach o podanych współrzędnych. Załóż, że standardowe wejście będzie zawierać  $n + 1$  linii, z których pierwsza będzie zawierać jedną liczbę całkowitą  $n$ , a pozostałe – po dwie liczby zmiennopozycyjne, stanowiące odpowiednio współrzędne  $x$  i  $y$  kolejnych wierzchołków łamanej. Na standardowe wyjście powinna być wypisywana jedna liczba zmiennopozycyjna z rezultatem obliczeń. Użyj funkcji `sqrt` z pliku nagłówkowego `math.h`.

**Zadanie 2 (10 pkt.)**

Napisz program do interaktywnego ćwiczenia dodawania i odejmowania liczb całkowitych. Program powinien:

- na początku jednorazowo pytać użytkownika o to, na ilocyfrowych liczbach chce ćwiczyć (odpowiedź powinna być dodatnią liczbą całkowitą);
- losować operację (dodawanie i odejmowanie) oraz argumenty; jeśli wylosowane zostało odejmowanie, to pierwszy argument powinien być nie mniejszy od drugiego (aby wynik nie był ujemny);
- wypisywać wyrażenie z wylosowanymi wartościami i oczekiwać na podanie wyniku przez użytkownika;
- jeśli użytkownik podał niepoprawny wynik, pytanie powinno zostać powtórzone; w przeciwnym wypadku należy wylosować nowe, itd.

Do losowania użyj funkcji `rand` z pliku nagłówkowego `stdlib.h` i reszty z dzielenia (operator `%`); żeby faktycznie zobaczyć (pseudo)losowe zachowanie, na początku programu (ale nie przed każdym wywołaniem funkcji `rand`!) umieść instrukcję `srand(time(NULL))`; (co z kolei będzie wymagać pliku nagłówkowego `time.h`). Możesz użyć pętli nieskończonej i wychodzić z programu skrótem klawiszowym `Ctrl+C`.

Z 10 pkt. za to zadanie 2 pkt. przyznawane będą za "user-friendliness", tj. czytelność komunikatów czy kontrolowane zachowanie, kiedy użytkownik wprowadzi dane niezgodne z oczekiwaniami (tj. inne niż liczby całkowite).

**Zadanie 3 (10 pkt., dla sprawdzaczki):**

Liczby Fibonacciego to ciąg zdefiniowany taką zależnością rekurencyjną (w niektórych źródłach podaje się czasem inne indeksy dla początkowych wyrazów, ale tu umawiamy się na poniższe):

$$F_0 = 0$$

$$F_1 = 1$$

$$F_{n+2} = F_{n+1} + F_n \text{ dla } n \geq 0$$

przy czym oczywiście lepiej wyliczać go iteracyjnie, a nie rekurencyjnie.

Przedstawienie liczby naturalnej  $n$  w układzie liczb Fibonacciego to taki ciąg  $a_2, a_3, \dots, a_k$  (zauważ brak  $a_0, a_1$ ) składający się z zer i jedynek, w którym nie występują dwie jedyne pod rząd,  $a_k = 1$ , oraz

$$a_2 F_2 + a_3 F_3 + \dots + a_k F_k = n.$$

W pierwszym i jedynym wierszu standardowego wejścia podana będzie jedna dodatnia liczba naturalna  $n \leq 2 \times 10^9$ . Napisz program, który ją wczyta, oraz wypisze jej przedstawienie w układzie liczb Fibonacciego począwszy od najbardziej znaczącego wyrazu, tj. od  $a_k$  do  $a_2$ . W jedynym wierszu standardowego wyjścia powinny znajdować się tylko jedynki i zera, bez spacji.

Program powinien czytać jedną liczbę całkowitą bez znaku ze standardowego wejścia używając funkcji `scanf`, np. `scanf("%u", &nazwa_zmiennej)`.

Każda liczba naturalna ma jednoznaczne przedstawienie w układzie liczb Fibonacciego, które można wyliczać zachłannie. (Będziesz musiał/-a to udowodnić dopiero na matematyce dyskretnej, tutaj wystarczy z tego skorzystać.)