Zajęcia 4

Temat: Pętla for, while, do-while 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, testuje poprawność programów dla różnych danych, posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów; bierze pod uwagę złożoność obliczeniową

Efekty: ,

- umie napisać program z wykorzystaniem instrukcji warunkowej i iteracyjnej,
- wie jak optymalizować program,
- umie policzyć złożoność obliczeniową

Formy i metody pracy: praca samodzielna, omówienie, wykład

Zadania do wykonania na zajęciach	Treści programowe
1. Dzielniki	M.2, P.2.10, A.2
2. Podzielne	M.2, P.2.10, A.2
3. Przenoszenie	M.2, P.2.10, A.2
4. Ścieżka Sterna-Brocota	M.2, P.2.10, A.2

Materiały do zajęć:

https://www.main2.edu.pl/main2/courses/show/6/14/

https://www.main2.edu.pl/main2/courses/show/6/17/ (część dot. pętli)

Zadania do wykonania w domu:

Choinka 2:
https://szkopul.edu.pl/problemset/problem/LNqt3rNp7kiR-65zBzZS3_sL/site/?key=statement

Wężyk:
https://szkopul.edu.pl/problemset/problem/T8JDkgJLZX_h9Hd_ead-sTum/site/?key=statement

ZADANIA I ROZWIĄZANIA

Zadanie 1. Dzielniki

Dostępna pamięć: 256MB. Źródło: szkopuł.edu.pl

Zadaniem Twojego programu będzie wypisanie wszystkich naturalnych dzielników zadanej liczby. Napisz program, który wczyta ze standardowego wejścia liczbę naturalną n, a następnie wypisze na standardowe wyjście wszystkie dzielniki liczby uporządkowane rosnąco.

Wejście

W pierwszej linii wejścia znajduje się jedna liczba całkowita n ($1 \le n \le 10^{12}$).

Wyjście

W i-tym wierszu wyjścia należy wypisać i-ty z kolei dzielnik liczby n.

Przykład

Wejście	Wyjście
12	1 2 3 4 6 12

Rozwiązanie

Należy zwrócić uwagę na szybkość oraz poprawność rozwiązania.

Rozwiązanie wolne sprawdza wszystkie dzielniki z zakresu [1, n]. Wymaga więc wykonania n operacji sprawdzenia podzielności.

```
dla i=1 do n wykonaj
      jeżeli n mod i = 0
            wypisz i
```

Rozwiązanie szybsze korzysta z właściwości: $a \cdot b = n$ dla $a \le \sqrt{n}$ i $b \ge \sqrt{n}$.

Stąd wystarczy sprawdzać wyłącznie liczby mniejsze bądź równe $\sqrt{n}$.

Ponadto funkcję matematyczną pierwiastek(n) można wyeliminować z poniższego zapisu podnosząc strony nierówności do kwadratu (rozwiązanie wzorcowe w C++).

```
dla i=1 do √n wykonaj
      jeżeli n mod i = 0
            wypisz i
dla i=√n do i wykonaj
      jeżeli n mod i = 0 ∧ i·i<n
            wypisz n div i
```

Zadanie 2. Podzielne

Dostępna pamięć: 32MB

Dane są liczby naturalne a i b. Wypisz, ile liczb w przedziale jest podzielnych przez 3 lub przez 5.

Wejście

W pierwszej linii wejścia znajdują się dwie liczby całkowite a i b ($1 \le a \le b \le 10^{15}$).

Wyjście

Na standardowym wyjściu należy wypisać jedną liczbę całkowitą.

Przykład

Wejście	Wyjście
25 75	24

Rozwiązanie

Należy zwrócić uwagę na szybkość oraz poprawność rozwiązania.

Rozwiązanie wolne sprawdza wszystkie liczby z zakresu [a, b]:

```
ile ← 0
dla i=a do b wykonaj
     jeżeli i mod 3 = 0 lub i mod 5 = 0
          ile ← ile + 1
wypisz ile
```

Wymaga więc wykonania b – a + 1 operacji sprawdzenia podzielności. W skrajnym wypadku będziemy być może musieli wykonać aż 10^{15} operacji. Zakładając, że komputer wykonuje około 10^8 operacji sprawdzenia podzielności w ciągu sekundy, potrzebowalibyśmy na to rozwiązanie około 10^7 sekund, co daje prawie 116 dni.

Zauważmy jednak, że liczb podzielnych przez 3 w przedziale [1, b] jest b div 3. A zatem w przedziale [a, b] liczb podzielnych przez 3 jest b div 3 – (a-1) div 3. Podobnie postępujemy dla 5. Pamiętajmy, że musimy wyeliminować liczby jednocześnie podzielne przez 3 i przez 5, które policzyliśmy dwukrotnie.

```
podzielne_przez_3 ← b div 3 - (a - 1) div 3
podzielne_przez_5 ← b div 5 - (a - 1) div 5
podzielne_przez_15 ← b div 15 - (a - 1) div 15
wypisz podzielne_przez_3 + podzielne_przez_5 - podzielne_przez_15
```

Ten program wykonuje tylko 3 operacje przypisania, działa więc w czasie stałym, niezależnym od liczb a oraz b.

Zadanie 3. Przenoszenie

Jasio uczy się dodawać wielocyfrowe liczby od prawej do lewej, po jednej cyfrze. Dla Jasia operacja przeniesienia, podczas której jedynka jest przenoszona z jednej pozycji do następnej, stanowi poważne wyzwanie. Twoim zadaniem jest policzenie, ile operacji przeniesienia wystąpi w każdym z dodawań w danym zestawie. Pomoże to Jasiowi w oszacowaniu trudności zadań.

Wejście

W pierwszej linii wejścia znajduje się liczba n ($1 \le n \le 25$) – liczba zestawów testowych. W każdym z n następnych wierszy znajdują się po dwie liczby całkowite bez znaku, każda z nich ma mniej niż 18 cyfr.

Wyjście

Dla każdego z n zestawów liczb wypisz liczbę operacji przeniesienia występujących podczas dodawania dwóch liczb.

Przykład

Wejście	Wyjście
3 234 342 654 456 191 111	0 3 1

Rozwiązanie

Zauważmy, że dla każdej z par liczb wystarczy sprawdzać kolejne cyfry począwszy od ostatniej do pierwszej dodając je do siebie wraz z bitem przeniesienia (na początku równym 0). Bit przeniesienia pojawia się wówczas, gdy suma dwóch ostatnich cyfr jest równa co najmniej 10. Wynikiem jest liczba bitów przeniesienia równych 1. Ponieważ liczby mogą składać się z różnej liczby cyfr, musimy liczyć względem większej z nich.

Zadanie dodatkowe dla uczniów: udowodnij, że bit przeniesienia nie będzie większy niż 1.

Operacja zliczania przeniesień dla dwóch liczb:

```
wczytaj a, b
ile ← 0
bit ← 0
dopóki a ≠ 0 lub b ≠ 0 wykonuj
      bit ← (bit + a mod 10 + b mod 10) div 10
      ile ← ile + bit
wypisz ile
```

Jak szybko działa nasze rozwiązanie? Pętla wykonuje liczbę „obrotów” zgodną liczbą cyfr większej liczb, jest więc równa $\lfloor \lg(\max(a,b)) \rfloor + 1$.

Zadanie 4. Ścieżka Sterna-Brocota

Drzewo Sterna-Brocota to drzewo binarne zawierające wszystkie dodatnie ułamki nieskracalne. Struktura ta posiada wiele ciekawych właściwości. Jeśli liczby a oraz b są względnie pierwsze, to ułamek $\frac{a}{b}$ występuje w drzewie dokładnie jeden raz. Ponadto każdą liczbę rzeczywistą dodatnią możemy zapisać jako ciąg symboli L oraz P tak, że początkowe fragmenty tego ciągu symbolizują liczby wymierne przybliżające tę liczbę. Na przykład liczbę $\frac{5}{7}$ opiszemy jako LPPL.

Zaczynamy od $\frac{0}{1}$ symbolizującego zero i $\frac{1}{0}$ symbolizującego nieskończoność. Następnie na kolejnych piętrach drzewa wpisujemy „pomiędzy” wartości $\frac{a}{b}$ oraz $\frac{c}{d}$ wartość $\frac{a+c}{b+d}$.

Naszymi wartościami startowymi jest $\frac{0}{1}, \frac{1}{1}, \frac{1}{0}$.

Zatem w pierwszym kroku mamy: $\frac{0}{1} \quad \frac{1}{1} \quad \frac{1}{0}$

W drugim kroku: $\frac{0}{1} \quad \frac{1}{2} \quad \frac{1}{1} \quad \frac{2}{1} \quad \frac{1}{0}$

W trzecim: $\frac{0}{1} \quad \frac{1}{3} \quad \frac{1}{2} \quad \frac{2}{3} \quad \frac{1}{1} \quad \frac{3}{2} \quad \frac{2}{1} \quad \frac{3}{1} \quad \frac{1}{0}$

Zaś w czwartym: $\frac{0}{1} \quad \frac{1}{4} \quad \frac{1}{3} \quad \frac{2}{5} \quad \frac{1}{2} \quad \frac{3}{5} \quad \frac{2}{3} \quad \frac{3}{4} \quad \frac{1}{1} \quad \frac{4}{3} \quad \frac{3}{2} \quad \frac{5}{3} \quad \frac{2}{1} \quad \frac{5}{2} \quad \frac{3}{1} \quad \frac{4}{1} \quad \frac{1}{0}$

Napisz program, który czyta dwie liczby m oraz n i wypisuje ścieżkę Sterna-Brocota.

Wejście

Jedyny wiersz danych zawiera dwie *względnie pierwsze* liczby całkowite naturalne m i n ($1 \le m, n \le 10^5$, $m \neq n$).

Wyjście

Program powinien wypisać ścieżkę z drzewa Sterna-Brocota.

Przykład

Wejście
`5 3`
Wyjście
`PLP`
	Wejście
	`7 2`
	Wyjście
	`PPPL`

Rozwiązanie

Zadanie wymaga zapamiętanie lewego i prawego zakresu $(\frac{0}{1}, \frac{1}{0})$, a następnie wyznaczania nowego „środka” i porównywanie go z $\frac{m}{n}$ (obliczona wartość może być mniejsza – P, większa – L, lub równa, co kończy nasze zadanie).

Wykorzystamy pętlę `do-while`:

```
a ← 0, b ← 1, c ← 1, d ← 0
wykonuj
      e ← a + c
      f ← b + d
      jeżeli m/n > e/f
            a ← e, b ← f
```

```
        wypisz 'P'
    jeżeli m/n < e/f
        c ← e, d ← f
        wypisz 'L'
dopóki m/n ≠ e/f
```

Aby wszystkie operacje wykonywać wyłącznie na liczbach rzeczywistych wykorzystamy:

$$\frac{\mathrm{x1}}{\mathrm{y1}} = \frac{x2}{\mathrm{y2}} \quad \equiv \quad x1 \cdot y2 = y1 \cdot x2 \qquad dla\ y1, y2 \geq 1$$

Zadanie dla uczniów: ile „obrotów" pętli wykona powyższy algorytm w skrajnym wypadku?