

## Lista zadań nr 3

Pętle: **while**, **do...while** i **for**.

### Zadania podstawowe:

**Zadanie 1** Napisz program, który prosi użytkownika o podanie liczby całkowitej, po czym wyświetla 10 kolejnych liczb całkowitych większych od wprowadzonej liczby. Wykorzystaj pętlę **while**. Zanim napiszesz kod programu w języku C zaprojektuj program za pomocą:

- schematu blokowego;
- pseudojęzyka.

**Zadanie 2** Napisz program, który liczy za użytkownika. Pozwól użytkownikowi na wprowadzenie liczby początkowej, liczby końcowej i wielkości odstepu między kolejnymi liczbami. Wykorzystaj pętlę **while**.

**Zadanie 3** Napisz program, który obliczy liczbę cyfr w podanej przez użytkownika liczbie naturalne (typ `unsigned long long`). Wykorzystaj pętlę **do...while**. Zanim napiszesz kod programu w języku C zaprojektuj program za pomocą:

- schematu blokowego;
- pseudojęzyka.

**Zadanie 4** Napisz program, który zliczy ile jedynek występuje w reprezentacji binarnej (typ `unsigned long long`) danej liczby naturalnej.

**Zadanie 5** Korzystając z pętli **while** wypisz:

- $n$  kolejnych liczb naturalnych począwszy od 1 (dla  $n > 1$  podanego przez użytkownika);
- Kolejne liczby naturalne od  $n$  do 1 (dla  $n > 1$  podanego przez użytkownika);
- wszystkie liczby naturalne parzyste i mniejsze od  $n$  (dla  $n > 1$  podanego przez użytkownika);
- $n$  kolejnych liczb nieparzystych począwszy od  $n$  (dla  $n > 1$  podanego przez użytkownika);
- $n$  kolejnych wyrazów ciągu arytmetycznego: 1, 4, 7, 10, 13, ... (dla  $n > 1$  podanego przez użytkownika);

- f) 12 kolejnych silni liczb naturalnych tj. 12 początkowych wyrazów ciągu:  $1, 1 \cdot 2, 1 \cdot 2 \cdot 3, 1 \cdot 2 \cdot 3 \cdot 4, \dots$ ;
- g)  $n$  kolejnych wyrazów ciągu:  $a_n = \frac{1}{n}$  (dla  $n > 1$  podanego przez użytkownika);
- h) 17 kolejnych wyrazów ciągu zdefiniowanego rekurencyjnie:  $a_1 = 3, a_n = 3(a_{n-1} - 1) - 4(3a_{n-1} - 4)$ ;
- i) 31 kolejnych wyrazów ciągu danego rekurencyjnie (pierwszym wyrazem jest  $a_0$ ):  $a_0 = 0, a_n = 2a_{n-1} + 1$  (dla  $n > 1$ )<sup>1</sup>; Czy Twój program potrafi poprawnie wyświetlić 32 wyraz tego ciągu? Jeżeli nie, to dlaczego jak można to naprawić?
- j) sumę  $n$  kolejnych liczb naturalnych tj. sumę  $1 + 2 + 3 + \dots + n$  (dla  $n > 1$  podanego przez użytkownika);
- k) sumę  $n$  kolejnych kwadratów liczb naturalnych tj. sumę  $1^2 + 2^2 + 3^2 + \dots + n^2$  (dla  $n > 1$  podanego przez użytkownika);
- l) ile jest liczb parzystych i nieparzystych w podanym przez użytkownika ciągu  $n$  liczb naturalnych (dla  $n > 1$  podanego przez użytkownika);
- m) ile jest liczb ujemnych i dodatnich w podanym przez użytkownika ciągu  $n$  liczb całkowitych (dla  $n > 1$  podanego przez użytkownika);

Wykonaj to samo zadanie używając pętli for.

**Zadanie 6** Napisz program szukający największej liczby w szeregu wprowadzonym przez użytkownika na wejściu. Program powinien pytać użytkownika o kolejne liczby. Kiedy użytkownik wprowadzi 0 albo liczbę ujemną, program powinien wyświetlić największą dotychczas wprowadzoną liczbę nieujemną. Zauważ, że liczby nie muszą być całkowite.

**Zadanie 7** Napisz program pytający użytkownika o dwie dodatnie liczby całkowite i obliczający największy wspólny dzielnik (NWD) tych liczb.

Wskazówka: Klasyczny algorytm NWD (tzw. algorytm Euklidesa) wygląda tak:

Niech  $m$  i  $n$  będą zmiennymi oznaczającymi dwie liczby całkowite. Jeżeli  $n$  jest zerem - koniec algorytmu,  $m$  jest NWD. W innym przypadku oblicz resztę z dzielenia  $m$  przez  $n$ . Skopiuj  $n$  do  $m$ , a resztę z dzielenia do  $n$ . Powtarzaj dopóki  $n$  nie jest zerem.

Zanim napiszesz kod programu w języku C zaprojektuj program za pomocą schematu blokowego.

---

<sup>1</sup>Podany ciąg jest rozwiązaniem tzw. *problemu wież z Hanoi*.

**Zadanie 8** Agata wpłaca na lokatę 1000 zł z oprocentowaniem wynoszącym 10% w skali roku (każdego roku lokata daje zysk w wysokości 10% jej początkowej wartości - procent prosty). Franek wpłaca również 1000 zł, ale na lokatę z kapitalizacją odsetek i oprocentowaniem równym 6% w skali roku (6% od bieżącej wartości, zawierającej w sobie doliczone odsetki z poprzedniego roku - procent składany). Napisz program, który obliczy po ilu latach wartość inwestycji Franek przekroczy wartość inwestycji Agaty, oraz wyświetli obie wartości w tym momencie. Wykorzystaj pętlę do...while.

**Zadanie 9** Napisz program który pobiera liczbę całkowitą ( $> 2$ ), po czym wyświetla wszystkie liczby pierwsze mniejsze bądź równe tej liczbie.

**Zadanie 10** Napisz program, który "zgaduje" liczbę pomyślaną przez użytkownika. Zakładamy, że liczba jest z przedziału od 1 do 100. Program powinien stosować strategię zgadywania zwaną przeszukiwaniem dwudzielnym. Polega ona na tym, że na przykład na początek program pyta użytkownika czy jego pomyślana liczba jest równa, większa czy mniejsza od 50. Jeżeli szukana liczba jest mniejsza od 50 to następną próbą powinno być 25, które znajduje się w połowie drogi między 1 a 50. Jeśli 25 jest zbyt małe to kolejną próbą powinna być liczba pomiędzy 25 a 50, i tak dalej. Przyjmij, że użytkownik nie oszukuje.

**Zadanie 11** Wykorzystując pętlę zagnieżdżoną napisz programy wyświetlające podane wzory, gdzie liczba wyświetlanych wierszy jest określana przez użytkownika:

a) \*

```
  **
 ***
****
*****
```

b) \*\*\*\*\*

```
 *****
 *****
  ****
   ***
    **
     *
```

c) \*

```
  ***
 *****
*****
*****
*****
```

## Zadania dodatkowe:

**Zadanie 1** Napisz program który służy do obliczania średniej ocen studentów. Użytkownik powinien podać liczbę studentów ( $n$ ) a następnie powinien wpisywać oceny kolejnych studentów. Użytkownik powinien mieć możliwość obliczenia średniej dla mniejszej liczby studentów niż  $n$  np. gdy użytkownik poda ocenę ujemną. Zastosuj instrukcję `break`.

**Zadanie 2** Napisz program, który dla pięciu małych liter podanych przez użytkownika wyświetli ich duże odpowiedniki. Program powinien ignorować znaki nie będące małymi literami - użyj instrukcji `continue`.

**Zadanie 3** Przypuśćmy, że chcemy obliczyć średnie zarobki pracowników w danej firmie dla tych, którzy rocznie zarabiają więcej niż 45000 zł, ale dysponujemy tylko ich miesięcznymi zarobkami brutto. Napisz program, który pyta o miesięczną pensję każdego pracownika, czyni z niej pensję roczną (mnożąc przez 12) i oblicza średnią. Jeżeli użytkownik wprowadzi wartość ujemną jako pensję miesięczną, program powinien zakończyć swoje działanie i drukować wartość średnią. Zastosuj pętlę nieskończoną i instrukcje `break` oraz `continue`.

**Zadanie 4** Napisz program, który oblicza sumę 1, 2, 3, 4 itd. (aż do pewnej granicy) wyrazów następujących ciągów:

$$\begin{aligned} &+1.0/2.0 + 1.0/3.0 + 1.0/4.0 + \dots \\ &-1.0/2.0 + 1.0/3.0 - 1.0/4.0 + \dots \end{aligned}$$

Maksymalna liczba wyrazów sumowania powinna być określana przez użytkownika. Przyjrzyj się sumom dla 10, 100 i 1000 wyrazów. Czy któryś z ciągów wydaje się zbiegać do jakiejś granicy.

Wskazówka:  $-1$  pomnożone przez siebie nieparzystą ilość razy jest równe  $-1$ , natomiast parzystą ilość razy  $+1$ .

**Zadanie 5** Napisz program, który wypisze liczby doskonałe z zadanego przedziału. Liczba doskonała to liczba, która jest równa sumie swoich wszystkich swych dzielników właściwych (to znaczy mniejszych od niej). Przykładowo liczbami doskonałymi są liczby 6 i 28 bo:  $6 = 1 + 2 + 3$  oraz  $28 = 1 + 2 + 4 + 7 + 14$ .

**Zadanie 6** Napisz program, który odczytuje dane wejściowe ze standardowego wejścia aż do wystąpienia znaku `#`, zastępując każdą kropkę wykrzyknikiem a każdy wykrzyknik - dwoma wykrzyknikami, oraz wyświetla liczbę zmian jaką wykonał. Napisz program używając

a) instrukcji `if ... else`,

b) instrukcji `switch`.

**Zadanie 7** Napisz program, który pobiera od użytkownika liczbę całkowitą i podaje polskie nazwy poszczególnych cyfr. Przykładowo dla liczby 734 program powinien wyświetlić napis:

siedem trzy cztery

Pamiętaj o wyświetleniu napisu zero gdy użytkownik wpisze liczbę 0. Wykorzystaj instrukcję `switch`.

**Zadanie 8** Napisz program obliczający sumę  $n$  składników:  $1 + 22 + 333 + 4444 + \dots$  (dla  $n > 1$  podanego przez użytkownika).

**Zadanie 9** Wykorzystując pętle zagnieżdżone napisz programy wyświetlające podane wzory, gdzie liczba wyświetlanych wierszy jest określana przez użytkownika:

a) E

ED

EDC

EDCB

EDCBA

b) A

BC

DEF

GHIJ

KLMNO

c)

A

ABA

ABCBA

ABCDcba

ABCDEDcba