

## Lista 1

W zadaniach dotyczących łańcuchów znaków, zdań lub ciągów podanych przez użytkownika zakładamy, że są one nie dłuższe niż 1024 znaki.

1. Napisz program, który pyta o imię użytkownika. Następnie zlicza ile liter 'a' (nie rozróżniamy dużych i małych liter) występuje w tym imieniu.
2. Napisz program, który stworzy linię z gwiazdek o długości zadanej przez użytkownika.
3. Napisz program, który stworzy trójkąt z gwiazdek na podstawie liczby podanej przez użytkownika. Trójkąt dla liczby 5 ma postać:

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

4. Napisz program, który wyświetla na ekranie "choinkę" o wysokości zadawanej z przez użytkownika. Dla liczby 4 choinka ma postać:

```
  *
 ***
*****
 #
-----
```

5. Napisz program, który rysuje na ekranie szachownicę o wymiarze  $n$  na  $n$ , składającą się ze znaku spacji i #. Liczbę  $n$  podaje użytkownik. Np., dla  $n=4$  mamy:

```
# #
 # #
# #
 # #
```

6. Napisz program z zadania poprzedniego z wykorzystaniem tylko jednej pętli.
7. Napisz program, który wypisuje na ekranie podane przez użytkownika słowo od końca (tzn. litery występują w odwrotnej kolejności).
8. Napisz program, który wypisuje na ekranie podane przez użytkownika zdanie dużymi literami z pominięciem spacji. Np. dla podanego zadania: Tomek ma 12 lat. program wypisze: TOMEKMA12LAT.
9. Napisz program, który sprawdza czy podany przez użytkownika ciąg znaków jest palindromem (tzn. czytany od końca brzmi tak jak oryginalny ciąg). Np.: kobyła ma mały bok
10. Napisz program, który zamienia w podanym przez użytkownika zdaniu słowo kotek na piesek. Np. dla podanego zdania: Ten kotek jest ładny. program wypisze: Ten piesek jest ładny.

**11.** Napisz program, który zlicza ilość wystąpień par sąsiadujących ze sobą liter 'o' w podanym przez użytkownika ciągu. Program nie rozróżnia małych i dużych liter. Np. dla podanego ciągu: `Ooo! This is a book.`  
program wypisze: `3`, ponieważ mamy pary:  
`Ooo! This is a book.`, `Ooo! This is a book.` oraz `Ooo! This is a book.`

**12.** Napisz program, który oblicza ilość słów w podanym przez użytkownika zdaniu.

**13.** Napisz program, który oblicza sumę cyfr liczby naturalnej podanej przez użytkownika.

**14.** Napisz program, który zapisuje podaną przez użytkownika liczbę naturalną w postaci binarnej.

**15.** Napisz program, który wyświetla na ekranie tabliczkę mnożenia o wymiarze zadanym przez użytkownika.

**16.** Napisz kalkulator. Program pyta o liczbę `a`, o rodzaj działania (czyli jeden ze znaków `+`, `-`, `*`, `/`), oraz o liczbę `b`. W przypadku dzielenia program sprawdza poprawność danych, aby nie wystąpiło dzielenie przez zero. Jeśli liczby były poprawne, to wypisuje wynik działania na liczbach `a` i `b` a następnie pyta:  
`Czy chcesz kontynuowac obliczenia? [t/n]`

**17.** Napisz program, który oblicza wszystkie (zespolone też) pierwiastki równania

$$ax^2 + bx + c = 0 .$$

Współczynniki są typu `double`.

Uwaga. Możesz skorzystać z funkcji `sqrt()` z biblioteki `math.h` .

**18.** Napisz program, który oblicza rozwiązania (jeśli istnieją) układu równań

$$\begin{cases} ax + by = c \\ dx + ey = f \end{cases} .$$

Współczynniki są typu `double`.

**19.** Napisz program, który zamienia podaną przez użytkownika liczbę całkowitą w postaci ciągu znaków na liczbę typu `int`. Program przekształca ciągi (dopóki mają sens jako liczby całkowite) w następujący sposób:

```
123 na 123
81kot na 81
-456.1 na -456
mysz12 na 0
napis na 0 .
```

**20.** Napisz program testujący czy podana przez użytkownika liczba naturalna jest liczbą pierwszą. Zastanów się, czy napisany przez Ciebie kod można jeszcze zoptymalizować?

**21.\*** Napisz program (z wykorzystaniem tylko jednej pętli), który generuje wszystkie słowa długości 4 nad alfabetem  $\{a, b, c\}$ . Litery mogą się powtarzać. Przykładowe słowa: `aaaa`, `abbc`, `baba`