

# Pętle

0. Jakie wartości wyświetli poniższy program:

```
#include <stdio.h>
int main() {
    int y, x=1, wynik = 0;
    while (x<=10) {
        y = x*x;
        printf("%d\n", y);
        wynik += y;
        ++x;
    }
    printf("wynik = %d\n", wynik);
    return 0;
}
```

Wpisz ten program, uruchom go i sprawdź czy dobrze odgadłeś wynik.

1. Napisz program, który wypisuje kolejne potęgi liczby 2 nie przekraczające 2020.

2. Napisz program wyświetlający poniższą szachownicę:

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

3. Napisz program, który wczyta dwie liczby całkowite  $a$  i  $b$  ( $a < b$ ) a następnie wypisze wszystkie liczby całkowite pomiędzy  $a$  i  $b$ .

4. Napisz program, który wyliczy sumę liczb naturalnych od 1 do liczby  $n$  podanej przez użytkownika. Kiedy użytkownik poda liczbę mniejszą niż 1, to program powinien wypisać stosowny komunikat o błędzie.

5. Napisz program, który wczyta ze *stdin* 6 liczb całkowitych do tablicy, a następnie wypisze zawartość tej tablicy od początku i od końca.

6. Napisz program, który wczyta ze *stdin* 6 liczb rzeczywistych do tablicy, a następnie wyszuka i wypisze największą i najmniejszą liczbę umieszczoną w tej tablicy.

7. Napisz program, który ze *stdin* wczyta 6 liczb rzeczywistych do tablicy, a następnie wypisze liczbę liczb dodatnich w niej zawartych.

8. Napisz program rozmieszczający poniższe wzory oddzielnie jeden pod drugim. Użyj pętli *for* do generowania wzorów. Do wypisywania gwiazdek **należy** użyć instrukcji *putchar('\*')* (powodującej wyświetlanie ich obok siebie). **Uwaga:** dwa ostatnie wzory wymagają, by każdy wiersz zaczynał się od odpowiedniej liczby spacji.

```

*          ****          ****          *
**         ***          ***          **
***        **           **           ***
****       *            *            ****

```

9. Kierowcy interesują się zużyciem paliwa do swoich samochodów. Napisz program, który będzie pobierał z pliku liczbę kilometrów z licznika samochodu, liczbę zatankowanych litrów paliwa i cenę za litr paliwa dla każdego tankowania.

Dane o tankowaniu powinny być wpisane w jednym wierszu, a wiersze zawierające tylko białe znaki lub rozpoczynające się dwoma znakami minus -- powinny być pomijane. Zakładamy, że za każdym razem tankujemy do pełna.

Przykładowy plik z danymi może wyglądać tak (plik *tankowania-2020.txt*):

```

-- wrzesień
10300 30.4 4.25
10645 25.7 4.45

-- październik
11020 15.4 4.25
11100  5.7 4.19

```

Po wczytaniu danych program powinien obliczyć i wypisać dla każdego tankowania (za wyjątkiem pierwszego): liczbę przejechanych kilometrów, liczbę litrów potrzebnych na przejechanie 100 km oraz koszt przejechania 100 km.

Po przetworzeniu wszystkich informacji, program powinien obliczyć i wypisać zbiorczą informację dla wszystkich tankowań.

Swój program zacznij od modyfikacji poniższego kodu (plik *tankowania.c*):

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    FILE *in_handle;
    char bufor[BUFSIZ];
    unsigned int ui;
    double x, y;
    in_handle = fopen(argv[1], "r");
    while (fgets(bufor, BUFSIZ, in_handle) != NULL) {
        sscanf(bufor, "%u %lf %lf", &ui, &x, &y);
        printf("--> %d %.2f %.2f\n", ui, x, y);
    }
    fclose(in_handle);
    return EXIT_SUCCESS;
}

```

Po skompilowaniu, program uruchamiamy w następujący sposób:

```

./tankowania tankowania-2020.txt

```