

Pętle repetycyjne i warunkowe

1. Przykład

Poniższy program wyznacza wartość $\ln(2)$ według wzoru:

$$\ln(2) \approx 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots$$

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int main()
{
    int i, znak;
    float suma;

    i = 1;
    znak = -1;
    suma = 0;

    while(i < 100)
    {
        znak = -znak;
        suma = suma + znak * 1 / i;
        i = i + 1;
    }

    printf("Przybliżone ln(2) = %f\nDokładne ln(2) = %f", suma, log(2));

    return EXIT_SUCCESS;
}
```

- Przepisz i uruchom program.
- Czy daje poprawne wyniki? Jeśli nie, to zaproponuj poprawki.

2. Zadanie

Przepisz program z zadania 1 (obliczanie $\ln(2)$) z użyciem pętli `do()`.

3. Zadanie

Przepisz program z zadania 1 $\ln(2)$ z użyciem pętli `for()`.

4. Zadanie

Napisz program, który po wprowadzeniu dwóch liczb:

- x – wartość argumentu,
- ϵ – dokładność obliczeń,

wyznacza wartość e^x (często oznaczaną $\exp(x)$) z dokładnością ϵ z wykorzystaniem zależności:

$$\exp(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!}.$$

Kolejne wartości wyrazów szeregu obliczaj na podstawie następnych stosując wzór: **wyraz = wyraz * x / i**, gdzie **i** przyjmuje kolejne wartości całkowitoliczbowe. Sumowanie wyrazów szeregu powinno odbywać się do momentu, aż kolejny okaże się mniejszy od **eps**.

Definiowane funkcji

5. Przykład

Rozwiąż ponownie poprzednie zadanie wykorzystując funkcję. Skorzystaj ze szkieletu aplikacji:

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

float mojeExp(float, float);

int main()
{
    float x, eps;

    printf("x = ");
    scanf("%f", &x);

    printf("\neps = ");
    scanf("%f", &eps);

    printf("\nPrzyblizona = %f\nDokladna = %f", mojeExp(x, eps), exp(x));

    return EXIT_SUCCESS;
}

float mojeExp(float x, float eps)
{
    // tresc funkcji zwracajacej wartosc typu float
}
```

6. Zadanie

Zdefiniuj funkcję, która dla zadanej liczby $a > 0$ wyznacza przybliżoną wartość jej pierwiastka korzystając ze wzoru:

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right) \quad n=0, 1, 2, \dots$$

Obliczenia zakończ, gdy $|x_{n+1} - x_n| < \varepsilon$. Liczba ε oznacza dokładność obliczeń (np. $\varepsilon = 0,001$, przyjmij $x_0 = 1$). Wykorzystaj funkcje do policzenia: $\sqrt{2}$, $\sqrt{5}$, $\sqrt{2a}$, $\sqrt{a+3}$. Wskazówka: wartość bezwzględną można wyznaczyć za pomocą funkcji **fabs()** z biblioteki **math.h**.

7. Zadanie

Napisz program, w którym zostanie zdefiniowana funkcja wyznaczająca k-ty element ciągu:

$$a_k = \frac{2(k+1)(k+2)}{3k}.$$

W programie wykorzystać tę funkcję do policzenia:

- n pierwszych elementów ciągu,

- sumy elementów ciągu o parzystych indeksach z zakresu $\langle 2, n \rangle$.

Zadania dodatkowe

1. Napisz program wczytujący współrzędne n punktów (dana liczba całkowita) i wyświetlający ile z nich leży w trzeciej ćwiartce układu współrzędnych. Sprawdzenie czy dany punkt leży w odpowiedniej ćwiartce zrealizuj w postaci funkcji.
2. Napisz program wyznaczający:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Zdefiniuj i wykorzystaj do obliczeń funkcję **silnia()**, która wyznacza $n!$ dla danego n .

3. Napisz funkcję o prototypie: **void pio(float r, float *p, float *v);** obliczającą pole powierzchni p i objętość kuli v o danym promieniu r . Parametry p i v przekazywane są przez wskaźnik. Sprawdź działanie funkcji wywołując ją w programie dla kilku przykładowych wartości. Kontroluj wartości parametrów przed i po wywołaniu funkcji. Sprawdź jak będzie działała funkcja o wszystkich parametrach przekazywanych przez wartość.