

Podstawy programowania

Pętle, proste funkcje

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

int main()
{
    int i, sign;
    float sum;

    i = 1;
    sign = -1;
    sum = 0;

    while(i < 100)
    {
        sign = -sign;
        sum = sum + sign * 1 / i;
        i = i + 1;
    }

    printf("Przybliżone ln(2) = %f.\nDokładne ln(2) = %f.\n", sum, log(2));

    return 0;
}
```

- 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 while()**.

3. Zadanie

Przepisz program z zadania 1 z użyciem pętli **for()**.

4. Zadanie

Napisz program, który po wprowadzeniu dwóch liczb:

- **x** – wartość argumentu,

- **eps** – dokładność obliczeń,

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

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

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

Definiowanie funkcji

5. Przykład

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

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

float myExp(float, float);

int main()
{
    float x, eps;

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

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

    printf("Przybliżona = %f.\nDokładna = %f.\n", myExp(x, eps), exp(x));

    return 0;
}

float myExp(float x, float eps)
{
    // treść funkcji zwracającej wartość typu float
}
```

6. Zadanie

Zdefiniuj funkcję, która dla zadanej liczby $a > 0$ wyznacza przybliżoną wartość jej pierwiastka korzystając z metody babilońskiej: $\sqrt{a} = \lim_{n \rightarrow \infty} x_n$, gdzie:

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

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: $\sqrt{2}$, $\sqrt{5}$, $\sqrt{2 \cdot a}$, $\sqrt{a+3}$. 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:

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

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:

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

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

3. Napisz funkcję o prototypie: **void sphere(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.