

Funkcje - wstęp

Deklaracja funkcji

`typ funkcji identyfikator_funkcji(lista parametrów formalnych);`

Typ funkcji określa typ wartości zwracanej przez funkcję (typ zdefiniowany pierwotnie jak `int`, typ zdefiniowany przez użytkownika np. klasa, lub typ `void`, który oznacza, że funkcja nie przekazuje wartości). Typem wartości zwracanych przez funkcję nie może być funkcja ani tablica. Lista parametrów formalnych składa się z typów argumentów oddzielonych przecinkami. Po każdym typie argumentu może nieobowiązkowo występować nazwa. Lista parametrów formalnych może być pusta, ale nie możemy opuścić nawiasów okrągłych: `int fun()` jest równoważne `int (void)`.

Napisz funkcję **SUMA** wyznaczającą sumę dwóch zadanych liczb całkowitych. Deklaracja funkcji **SUMA** może wyglądać w następujący sposób:

`int SUMA(int, int);`

lub

`int SUMA(int a, int b);`

Definicja funkcji

Definicja funkcji różni się od deklaracji. Składa się z nagłówka funkcji oraz treści funkcji ujętej w nawiasy klamrowe. W przypadku definicji wiersza nagłówkowego nie kończymy średnikiem. W treści funkcji, która ma określony typ wyniku, powinna wystąpić przynajmniej jedna instrukcja **return wyrażenie**;

Wartość tego wyrażenia będzie wynikiem funkcji. Gdy funkcja jest typu **void**, powinno wystąpić samo słowo **return**; Wykonanie instrukcji powoduje zakończenie wykonywania funkcji.

Definicja funkcji **SUMA**:

```
int SUMA(int a, int b)
{
    return a+b;
}
```

lub

```
int SUMA(int a, int b)
{
    int c=a+b;
    return c;
}
```

Wykorzystanie funkcji w programie.

W programie wykorzystujemy funkcję pisząc `identyfikator_funkcji(lista paramerów aktualnych)`. Nazwy parametrów aktualnych mogą różnić się od nazw parametrów formalnych. Kompilator sprawdza zgodność typów parametrów aktualnych z typami parametrów formalnych. Zgodne są typy identyczne oraz takie, dla których można dokonać automatycznej konwersji wartości.

```
#include<iostream>
using namespace std;
int main()
{
    int c1,c2;
    cout<<"c1=";
    cin>>c1;
    cout<<"c2=";
    cin>>c2;
    cout<<"c1+c2="<<SUMA(c1,c2)<<endl;
    system("pause");
    return 0;
}
```

Sposoby przekazywania parametrów.

Parametry do funkcji można przekazywać na trzy sposoby: przez wartość, przez referencję oraz przez wskaźnik. Dwa pierwsze sposoby zostaną omówione poniżej. Przekazywanie parametru przez wartość zostało zastosowane w definicji funkcji **SUMA**. Parametry przekazane w ten sposób nie zmieniają wartości pod wpływem działania funkcji (ponieważ są one lokalnymi kopiami tych argumentów i znajdują się na stosie podczas wykonywania funkcji).

Przekazanie parametru przez referencję odbywa się przez postawienie znaku **&** przed nazwą parametru i umożliwia zmianę wartości zmiennej, która jest parametrem aktualnym, pod wpływem działania funkcji (ponieważ funkcja otrzymuje adres argumentu). Z przekazywania parametrów przez referencję korzystamy gdy chcemy zmieniać wartości argumentów aktualnych lub chcemy przekazać dodatkowe wyniki funkcji do funkcji wywołującej.

Jeżeli parametrem funkcji jest tablica, jest ona przekazywana przez wskaźnik do pierwszego elementu tablicy, co umożliwia również zmianę wartości elementów tablicy pod wpływem działania funkcji. Wymiar tablicy należy przekazać jako osobny argument. (np. `float max(float tab[],int wymiar)`)

```
void ZwiekszOPiec1(int a)
{
    a+=5;
    return;
}
void ZwiekszOPiec2(int &a)
{
    a+=5;
    return;
}
```

Do funkcji **ZwiekszOPiec1** parametr został przekazany przez wartość, do funkcji **ZwiekszOPiec2** – przez referencję.

Wykorzystanie obu funkcji

```
#include<iostream>
using namespace std;
int main()
{
    int liczba=11;
    cout<<"Liczba ma wartość "<<liczba<<endl;
    ZwiekszOPiec(liczba);
    cout<<"Wartość zmiennej liczba po wykonaniu funkcji ZwiekszOPiec1 wynosi "
    <<liczba<<endl;
    //funkcja nie zmieniła wartości zmiennej liczba,
    //ze względu na przekazanie jej przez wartość
    ZwiekszOPiec2(liczba);
    cout<<"Wartość zmiennej liczba po wykonaniu funkcji ZwiekszOPiec2 wynosi "
    <<liczba<<endl;
    //funkcja zwiększyła wartość zmiennej. Teraz liczba=16
    system("pause");
    return 0;
}
```

Funkcje ćwiczenia

1. Napisz funkcję **ILOCZYN** wyznaczającą iloczyn dwóch zadanych liczb całkowitych. Działanie funkcji sprawdź pisząc odpowiedni program
2. Napisz funkcję **ILORAZ** wyznaczającą iloraz dwóch zadanych liczb całkowitych. Działanie funkcji sprawdź pisząc odpowiedni program.
3. Napisz funkcję, która wyznacza liczbę cyfr zadanej liczby całkowitej.

4. Napisz funkcję, która wyznacza n -tą (n jest liczbą naturalną) potęgę zadanej liczby rzeczywistej **x**. Funkcja ma wykorzystywać iterację.
5. Napisz funkcję, która wyznacza wartość $n!$ (n jest liczbą naturalną). Funkcja ma wykorzystywać iterację.
6. Napisz funkcję obliczającą euklidesową odległość pomiędzy dwoma punktami. Współrzędne punktów mają być zadane jako parametry.
7. Napisz funkcję zamieniającą wartościami liczby podane jako jej parametry.
8. Napisz funkcję **PoleIObwod**, która dla zadanych parametrami boków trójkąta zwróci jako wartość pole trójkąta i jako parametr zwróci obwód trójkąta. Jeśli parametry nie mogą być długościami boków trójkąta, funkcja ma zwracać wartość **-1**.
9. Napisz funkcję **SumaIIlosc**, która dla zadanej parametrem liczby całkowitej jako wartość zwróci sumę cyfr tej liczby i jako parametr zwróci ilość jej cyfr.
10. Napisz funkcję, której danymi są liczby oznaczające długość, szerokość i wysokość prostopadłościanu a wynikiem jest pole powierzchni ścian oraz objętość, przekazana jako parametr. Jeżeli któraś z wartości nie jest dodatnia, funkcja ma zwracać wartość **-1**.
11. Napisz program, który podaną przez użytkownika kwotę pieniędzy (liczba całkowita) rozmieni na jak najmniejszą ilość monet i banknotów o nominałach 1, 2, 5, 10 złotych. Przykład dla kwoty 188 zł:
188 zł rozmieniamy na
18 banknotów 10 zł
1 moneta 5 zł
1 moneta 2 zł
1 moneta x 1 zł.
W programie należy zdefiniować funkcję, w której zadana kwota oraz ilości poszczególnych monet są parametrami funkcji, a wynikiem funkcji jest ilość banknotów dziesięciozłotowych.
12. Dokończ definicję funkcji

```
int liczbawystapien(int n, int c)
```

która zwraca w wyniku informację ile razy cyfra **c** wystąpiła w zapisie dziesiętnym liczby **n**.
13. Napisz funkcję **max2**, która zwraca większą z dwóch podanych jako jej parametry liczb, a następnie funkcję **max3**, zwracającą największą spośród trzech podanych liczb. Funkcja **max3** ma do tego celu używać funkcji **max2**.
14. Napisz funkcję, która dla zadanych liczb rzeczywistych **a**, **b**, **c** zwraca **-1** jeśli nie mogą być to długości boków trójkąta, lub pole trójkąta o bokach o długości **a**, **b**, **c**. (Ta funkcja z kolei, może używać funkcji **max3**)
15. Napisz funkcję która ma trzy parametry formalne **a**, **b**, **c** będące liczbami całkowitymi. Wartością funkcji jest **true**, jeśli zadane liczby są liczbami pitagorejskimi oraz **false** w przeciwnym wypadku. Liczby pitagorejskie spełniają warunek: $a^2 + b^2 = c^2$. Wykorzystując tę funkcję, wydrukuj na ekranie wszystkie **różne** trójki pitagorejskie w przedziale od **<1, 30>**.
16. Napisz funkcję, która wyznacza pierwiastek całkowity z zadanej liczby. Zastosuj algorytm odejmowania kolejnych liczb nieparzystych (pierwiastek całkowity z liczby x związany jest z **ilością** kolejnych liczb nieparzystych jakie uda się odjąć od liczby x tak by pozostała ona nieujemna).
17. Napisz funkcję, która wyznacza sumę cyfr zadanej liczby całkowitej. Wykorzystaj tę funkcję w programie wyznaczającym dla dowolnej liczby całkowitej liczbę **jednocyfrową** powstałą przez sumowanie jej cyfr (jeśli suma cyfr nie jest liczbą jednocyfrową, to jej cyfry po raz kolejny sumujemy). Wydruk ma mieć postać:

Liczba podana przez uzytkownika: 2982
FSuma (2982) =FSuma (21) =3

18. Napisz funkcję logiczną, która stwierdza, czy zadana jako parametr liczba całkowita jest liczbą pierwszą. Wartością funkcji ma być prawda, jeśli parametr jest liczbą pierwszą oraz fałsz w przeciwnym wypadku.
19. Napisz funkcję logiczną **Doskonała**, która zwraca **true**, gdy zadana parametrem liczba całkowita jest liczbą doskonałą, **false** w przeciwnym przypadku. Liczba doskonała to taka, która jest równa sumie swoich dzielników. Przykładami takich liczb są $1 = 1$, $6 = 1+2+3$.
20. Ciąg liczb rzeczywistych jest określony wzorem
- $$a_0 = 2;$$
- $$a_n = 1 / (2 + a_{n-1}), \text{ dla } n \geq 1$$
- Napisz funkcję, która dla danej liczby naturalnej **n** oblicza n-ty wyraz ciągu a_n .
21. Ciąg $F_0=0$, $F_1=1$, $F_n=F_{n-1}+F_{n-2}$ nazywamy ciągiem Fibbonaciego. Napisz funkcję obliczającą wartość F_n dla danego **n**.
22. Wielomiany Czebyszewa pierwszego rodzaju określamy za pomocą wzorów $T_0(x)=1$, $T_1(x)=x$, $T_n(x)=2xT_{n-1}(x)-T_{n-2}(x)$. Napisz funkcję obliczającą wartość $T_n(x)$ dla danego **n** i punktu **x**.
23. Napisać funkcję obliczającą n-te przybliżenie liczby **e**, korzystając z rozwinięcia: $e = \sum_{k=0}^n \frac{1}{k!}$.
24. Napisać funkcję obliczającą n-te przybliżenie liczby e^{-1} , korzystając z rozwinięcia: $e^{-1} = \sum_{k=0}^n \frac{(-1)^k}{k!}$. (Wskazówka: 1sp. Skorzystaj ze wzorów rekurencyjnych $a_0=1$, $a_k=-a_{k-1}/k$, $k=1, 2, \dots, n$ oraz algorytmu sumowania, 2sp. skorzystaj ze schematu Hornera.)

Funkcje operujące na tablicach

25. Napisz funkcję, która znajduje w tablicy element maksymalny. Parametrami funkcji mają być: tablica **tab** oraz liczba elementów tablicy – **n**. Wartością funkcji ma być element maksymalny.
26. Napisz funkcję, która znajduje w tablicy element minimalny. Parametrami funkcji mają być: tablica **tab** oraz liczba elementów tablicy – **n**. Wartością funkcji ma być element minimalny.
27. Napisz funkcję, która posiada trzy parametry formalne. Pierwszym parametrem jest liczba **a**, drugim tablica liczb **tab** zaś trzecim – liczba **n**. Funkcja ma dla zadanej liczby **a** zwracać w tablicy **tab** jej kod binarny. Liczba **n** ma informować o tym ile elementów tablicy zostało wypełnionych.
28. Napisz funkcję, która sortuje zadaną tablicę liczb całkowitych malejąco.
29. Napisz funkcję, która sortuje zadaną tablicę liczb całkowitych rosnąco.
30. Napisz funkcję, która wyznacza wartość wielomianu w zadanym punkcie. Parametrami funkcji mają być: **n** - stopień wielomianu (liczba całkowita), **tab** – tablica współczynników (współczynniki są liczbami rzeczywistymi) oraz **x** – zadany punkt (liczba rzeczywista). Wartością funkcji ma być wyznaczona wartość wielomianu.
31. Napisz funkcję, która wyznacza sumę dwóch wielomianów.
32. Napisz funkcję, zwracającą liczbę elementów w tablicy większych od podanej liczby.
33. Napisz funkcję, zwracającą średnią arytmetyczną wszystkich elementów w tablicy.
34. Napisz funkcję, zwracającą iloczyn skalarny dwóch wektorów tego samego wymiaru.