

Lista zadań nr 4

Funkcje w języku C. Wskaźniki i funkcje operujące na wskaźnikach.

Zadania podstawowe:

Zadanie 1 Średnią harmoniczną dwóch liczb uzyskujemy przez znalezienie odwrotności danych liczb, obliczeniu z nich średniej arytmetycznej i wyliczeniu odwrotności otrzymanego wyniku. Zaprojektuj funkcję, która pobiera dwa argumenty typu `double` i zwraca ich średnią harmoniczną. Przetestuj napisaną funkcję w prostym programie.

Zadanie 2 Zaprojektuj funkcję `max(x, y)`, zwracającą większą z dwóch wartości typu `double`. Przetestuj napisaną funkcję w prostym programie.

Zadanie 3 Napisz iteracyjną wersję funkcji wyznaczającej wartość $n!$ (silnia) dla danego argumentu n . Korzystając z funkcji napisz program:

- a) wyliczający wartość wyrażenia $4! + 6! - 2!$;
- b) wyznaczający wartość symbolu Newtona $\binom{n}{k}$ dla $n = 5$ i $k = 2$.

Zadanie 4 Zaprojektuj funkcję `power()`, która zwraca wynik podniesienia liczby do potęgi całkowitej (wykładnik potęgi (`int`) i liczba (`double`) są argumentami tej funkcji). Użyj pętli. Przetestuj napisaną funkcję w prostym programie obliczając wartość wyrażenia $(3.1)^2 + (5.5)^3 + 24$.

Zadanie 5 Napisz funkcję, która jako argument otrzyma liczbę całkowitą dodatnią (typ `unsigned int` i policzy ile jedynek znajduje się w jej reprezentacji binarnej oraz zwróci tę wartość do miejsca wywołania. Przetestuj funkcję w prostym programie.

Zadanie 6 Napisz funkcję pobierającą trzy argumenty: znak i dwie liczby całkowite (bez znaku). Pierwsza liczba określa liczbę razy, jaką należy wyświetlić znak w jednym wierszu; druga określa liczbę wierszy. Funkcja powinna wyświetlać odpowiednią liczbę znaków w odpowiedniej ilości wierszy. Przetestuj napisaną funkcję w prostym programie.

Zadanie 7 Zaprojektuj funkcję `get_number()`, która będzie służyła do pobierania liczby całkowitej z podanego zakresu i zwracała ją. Funkcja powinna tak działać, aby "zmusić" użytkownika do podania liczby całkowitej z podanego zakresu (zakres zdefiniowany przez dwa argumenty funkcji). Przetestuj napisaną funkcję w prostym programie.

Zadanie 8 Napisz krótki program, w którym są zadeklarowane i zainicjalizowane cztery zmienne: znakowa, całkowita, zmiennopozycyjna i zmiennopozycyjna podwójnej precyzji. Niech program wyświetla wartość każdej zmiennej oraz jej adres w pamięci. Napisz dwie wersje programu:

- program wyświetla wartość każdej zmiennej oraz jej adres przy użyciu jej nazwy,
- w części programu, realizującej wyświetlanie ww. danych, nie jest używana nazwa danej zmiennej.

Zadanie 9 Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zwraca jako wartość mniejszą z liczb wskazywanych przez argumenty. Przetestuj napisaną funkcję w prostym programie.

Zadanie 10 Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `float`, która zwraca jako wartość sumę modułów z liczb wskazywanych przez argumenty. Przetestuj napisaną funkcję w prostym programie.

Zadanie 11 Napisz funkcję otrzymującą jako argumenty wskaźniki do dwóch zmiennych typu `int`, która zamienia ze sobą wartości wskazywanych zmiennych tylko wtedy, gdy wskazywana przez drugi argument zmienna jest mniejsza od zmiennej wskazywanej przez pierwszy argument. Przetestuj napisaną funkcję w prostym programie.

Zadanie 12 Zaprojektuj i napisz funkcję `bigger_number()`, która zamienia zawartość obu zmiennych typu `double` większą z nich. Wywołanie `bigger_number(a,b)` przypisze obu zmiennym `a` i `b` wartość większą z nich. Przetestuj funkcję w prostym programie.

Zadanie 13 Napisz funkcję, której argumentami są `n` typu `int` oraz `w` - wskaźnik do `int`, która przepisuje wartość `n` do zmiennej wskazywanej przez `w`. Przetestuj funkcję w prostym programie.

Zadanie 14 Zaprojektuj, napisz i przetestuj funkcję, która dokona dekompozycji liczby zmiennoprzecinkowej (`double`) na część ułamkową i część całkowitą (zapisane w dwóch osobnych zmiennych). Przetestuj funkcję w prostym programie.

Zadanie 15 Napisz funkcję typu `void`, która pobiera od użytkownika liczbę całkowitą z określonego zakresu (zakres zdefiniowany przez dwa argumenty funkcji). Przetestuj funkcję w prostym programie.

Zadania dodatkowe:

Zadanie 1 Napisz program, który pobiera dwie liczby zmiennoprzecinkowe (typy `float`) i wyświetla wartość ich różnicy podzielonej przez ich iloczyn. Program powinien odczytywać kolejne pary danych wejściowych aż do momentu, gdy użytkownik wpisze wartość nie będącą liczbą. Zaprojektuj program tak, aby obliczaniem wyniku zajmowała się osobna funkcja.

Zadanie 2 Napisz funkcję, która otrzymując jako argument liczbę całkowitą dodatnią (`unsigned long`) wyznaczy sumę cyfr tej liczby i ją zwróci do miejsca wywołania. Przetestuj napisaną funkcję w prostym programie.

Zadanie 3 Zdefiniuj funkcję `sum` obliczającą i zwracającą dla zadanego parametru n ($n = 1, 2, \dots, 9$) sumę składników: $1 + 22 + 333 + 4444 + \dots$. Przetestuj funkcję `sum` w prostym programie.

Zadanie 4 Napisz program, który oblicza pole powierzchni stożka. Funkcja `main()` powinna zawierać deklaracje dwóch zmiennych typu `double`: `height`, `radius`. Zaprojektuj i wykorzystaj w programie funkcję `get_data()`, która pobierze od użytkownika wartości zmiennych `height` i `radius` – funkcja jako argumenty ma przyjmować adresy tych zmiennych. Funkcja powinna korzystać z funkcji `get_double_plus()` w celu pobrania odpowiednich danych. Obliczaniem długości tworzącej stożka i jego pola powierzchni powinna zajmować się osobna funkcja, pobierająca jako argumenty adresy potrzebnych zmiennych i zwracająca wyliczone pole (wartość zwracana powinna być typu `double`). Skorzystaj z pliku nagłówkowego `math.h` zawierającego definicję stałej `M_PI` i prototyp funkcji `sqrt()` zwracającej pierwiastek z danej liczby.

Zadanie 5 Napisz program, który odczytuje znaki ze standardowego wejścia aż do wystąpienia znaku końca pliku (EOF). Dla każdego znaku program powinien informować, czy jest on literą i jeśli tak jest to program powinien wyświetlać numer litery w alfabecie. Na przykład, litery `d` i `D` mają numer 4. Wykorzystaj funkcję, która pobiera znak jako argument i zwraca jego numer w alfabecie, jeśli jest on literą; w przeciwnym wypadku wartością zwracaną powinni być `-1`.