

Zestaw 39

1. W folderze Debug39 znajduje się projekt z kodem w języku C. W pliku main.c w niektórych liniach są komentarze. Twoim zadaniem jest wpisanie wartości odpowiednich zmiennych po wykonaniu konkretnej linii kodu.

Punktacja: 1 pkt za każdą poprawną liniijkę, łącznie 7 pkt.

2. W folderze Popraw39 znajduje się kod w języku C, który nie spełnia zasad kompilacji. Popraw kod modyfikując dokładnie 1 liniijkę tak, aby się kompilował. Zabronioną operacją jest komentowanie kodu. Do zmodyfikowanych linii zaliczają się zarówno linie istotne ze względu na kompilację jak i te nieistotne (np. dodanie spacji przed operatorem może być operacją nieistotną ze względu na kompilację, ale będzie liczone jako zmodyfikowana liniijka).

Punktacja: 7 pkt.

3. Napisz funkcję, której parametrami są trzy dodatnie liczby całkowite n , m i k . Funkcja ma zwrócić iloczyn liczb większych niż n , mniejszych lub równych m i niepodzielnych przez k . Stwórz dwa przypadki testowe dla funkcji.

Punktacja: 8 pkt.

4. Napisz funkcję, która otrzymuje trzy argumenty:

- dwa wskaźniki na funkcje o jednym argumencie typu `double` zwracające wartość typu `double`,
- wartość `y` typu `double`.

Funkcja ma zwrócić 1 jeśli wartość pierwszej funkcji jest odwrotnością wartości drugiej funkcji odpowiednio na liczbach od 1 do n (np. $f_1(1) = \frac{1}{f_2(1)}$, $f_1(2) = \frac{1}{f_2(2)}$, itd.) oraz ma zwracać -1 w przeciwnym wypadku. Stwórz dwa przypadki testowe dla funkcji, aby funkcja zwróciła obie możliwe wartości.

Punktacja: 8 pkt.

5. Napisz funkcję rekurencyjną, która dla otrzymanej w argumencie nieujemnej liczby całkowitej n zwraca wartość elementu o indeksie n ciągu zdefiniowanego w następujący sposób

$$a_0 = 1, a_1 = 2$$

$$a_n = \begin{cases} a_{n-1} + 2, & n - \text{parzyste} \\ a_{n-2} * 2, & n - \text{nieparzyste} \end{cases}$$

Stwórz dwa przypadki testowe dla funkcji.

Punktacja: 10 pkt.

6. Napisz funkcję, która otrzymuje trzy argumenty: liczbę całkowitą n , dodatnią liczbę całkowitą m oraz m -elementową tablicę `tab` o elementach typu `int`. Funkcja ma zwrócić liczbę ile liczb mniejszych lub równych $2n$ jest w tablicy `tab`. Stwórz dwa przypadki testowe dla funkcji.

Punktacja: 10 pkt.