

Programowanie strukturalne - Kolokwium 2 - Zestaw B74

Rozwiązania mają być umieszczone zgodnie ze specyfikacją:

- Zadania powinny być umieszczone w archiwum .zip na udostępnionym pendrive.
- Nazwa archiwum powinna być wg schematu NUMERZESTAWU_NUMERALBUMU.zip gdzie numer zestawu znajduje się na górze kartki z poleceniami. np. A23_123456.zip
- We wnętrzu archiwum powinny znajdować się tylko same kody w języku C, pliki powinny posiadać dokładnie nazwy (z uwzględnieniem wielkości znaków): `zad1.c`, `zad2.c`, `zad3.c`, `zad4.c`, `zad5.c`.
- Maksymalna waga archiwum 10 MB.
- Archiwum powinno być bez hasła.
- W przypadku pominięcia danego zadania, należy dodać plik o nazwie sprecyzowanej wyżej (zawartość może być pusta).

Za zachowanie specyfikacji dokładnie otrzymuję się dodatkowe 2 punkty. Zadania znacznie odbiegające od specyfikacji mogą nie być sprawdzane.

1. W folderze DebugXYZ (XYZ - losowe znaki) znajduje się projekt z kodem w języku C. W pliku `main.c` w niektórych liniijkach są komentarze. Twoim zadaniem jest wpisanie wartości odpowiednich zmiennych po wykonaniu konkretnej linii kodu. Dopisanie nowych liniijek czy zaburzenie struktury kodu oznacza zero punktów za polecenie. W przypadku znaków, należy zapisać sam znak w apostrofach np. `'c'` (wielkość znaków ma znaczenie).

Punktacja: 6 pkt.

2. Napisz funkcję, której argumentami są dwa napisy. Funkcja powinna zwrócić informację ile znaków będących literami (bez względu na wielkość) jest w dłuższym napisie. Jeśli napisy są równej długości, to funkcja powinna zwrócić liczbę liter z drugiego napisu. Stwórz przypadek testowy. W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia.

Punktacja: 10 pkt.

3. Napisz funkcję, której argumentem jest dwuwymiarowa tablica elementów (zawierająca zmienne typu `int`) oraz jej wymiary n i m . Funkcja ma przestawić pierwszy i ostatni wiersz między sobą. Jeśli tablica ma mniej niż dwa wiersze, funkcja ma nic nie robić. Stwórz przypadek testowy.

Przykład:

$$\begin{bmatrix} 2 & 3 & -3 \\ 1 & 4 & 7 \\ -3 & -6 & 11 \\ -2 & 8 & 23 \end{bmatrix} \rightarrow \begin{bmatrix} -2 & 8 & 23 \\ 1 & 4 & 7 \\ -3 & -6 & 11 \\ 2 & 3 & -3 \end{bmatrix}$$

Punktacja: 10 pkt.

4. Napisz strukturę `Pralka` z polami `model` (tablica znaków długości 30) oraz `licznikPrania` (typu `int`). Następnie napisz funkcję `zrobPranie`, której argumentem jest wskaźnik do struktury typu `Pralka`. Funkcja ma dodać 1 do pola `licznikPrania` w przekazanym argumentcie. Stwórz przypadek testowy.

Punktacja: 12 pkt.

5. Napisz funkcję, która przyjmuje jako argument listę z głową o elementach typu:

```
struct element {
    int value;
    struct element * next;
};
```

oraz liczbę całkowitą `a`. Funkcja wyświetlić te elementy listy, które są większa niż `a`. W przypadku braku takich elementów, funkcja ma nic nie robić. Stwórz jeden przypadek testowy.

Punktacja: 22 pkt.