

Programowanie strukturalne - Kolokwium 2 - Zestaw B53

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 liniach są komentarze. Twoim zadaniem jest wpisanie wartości odpowiednich zmiennych po wykonaniu konkretnej linii kodu. Dopisanie nowych linii 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 argumentem jest napis. Funkcja ma zwrócić ile znaków będących przecinkami jest w napisie. Stwórz przypadek testowy.

Punktacja: 10 pkt.

3. Napisz funkcję, której argumentem jest dwuwymiarowa kwadratowa tablica tablic (zawierająca zmienne typu `int`) oraz jej wymiar n . Funkcja ma odwrócić kolejność elementów w wierszach o parzystych indeksach. Stwórz przypadek testowy.

Przykład:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 2 & 1 \\ 5 & 6 & 7 & 8 \\ 12 & 11 & 10 & 9 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Punktacja: 10 pkt.

4. Stwórz strukturę `Samochod` o dwóch polach `marka` (napis) oraz `cena` (dowolny typ całkowity). Następnie stwórz funkcję, której argumentami jest niepusta tablica struktur `Samochod` oraz rozmiar tablicy. Funkcja ma zwrócić najdroższą cenę za samochód z tablicy. Stwórz przypadek testowy.

Punktacja: 12 pkt.

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

```
struct elem {
    float x;
    struct elem * next;
};
```

Funkcja ma zwiększyć o 5 każdy element znajdujący się na liście (dodać do pola `x` wartość 5). Stwórz przypadek testowy.

Punktacja: 22 pkt.