

Programowanie strukturalne - Kolokwium 2 - Zestaw B73

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 argumentami są dwa napisy. Funkcja powinna zwrócić informację ile znaków będących małymi literami jest w krótszym napisie. Jeśli napisy są równej długości, to funkcja powinna zwrócić liczbę małych liter z pierwszego 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 kwadratowa tablica tablic (zawierająca zmienne typu `int`) oraz jej wymiar n , $n > 1$. Funkcja powinna zwrócić największy element znajdujący się na przekątnej pobocznej (suma indeksów jest równa $n-1$). Stwórz przypadek testowy.

Punktacja: 10 pkt.

4. Napisz strukturę `Komorka` z polami `numer` (tablica znaków długości 15) oraz `stanBaterii` (typu `int`). Następnie napisz funkcję `ladujBaterie`, której argumentem jest wskaźnik do struktury typu `Komorka`. Funkcja ma dodać 20 do pola `stanBaterii` w przekazanym argumencie. Jeżeli po dodaniu wartość przekracza 100, stan baterii powinien być ustawiony na 100. Stwórz przypadek testowy.

Punktacja: 12 pkt.

5. Napisz funkcję, która przyjmuje jako argument listę bez głowy o elementach typu:

```
struct node {  
    int val;  
    struct node * next;  
};
```

Funkcja ma wyświetlić w kolejnych wierszach te elementy listy, które są mniejsze niż pierwszy element. W przypadku gdy lista ma mniej niż dwa elementy, to funkcja ma nic nie wyświetlać. Stwórz przypadek testowy.

Punktacja: 22 pkt.