

Programowanie strukturalne

- Przykładowe Kolokwium 2

- Zestaw W01

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 oraz znak. Funkcja ma zwrócić ile razy znak występuje 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 zwrócić indeks wiersza, w którym znajduje się największy element - w przypadku kilku takich wierszy, zwróć indeks ostatniego z nich. Stwórz przypadek testowy.

Punktacja: 10 pkt.

4. Stwórz unię XYZ przechowującą dwa pola: `a` typu `int` i `b` typu `char`. Stwórz program zawierający tablicę 6 elementów typu XYZ. Wypisz na konsoli zawartość tablicy.

Punktacja: 12 pkt.

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

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

Funkcja ma usunąć pierwszy element na liście (o ile lista jest niepusta). Stwórz przypadek testowy.

Punktacja: 22 pkt.