Egzamin (pierwszy termin 2023) - Programowanie strukturalne - Zestaw E35

Zadanie 1: 6 pkt. Zadanie 2: 12 pkt. Zadanie 3: 14 pkt. Zadanie 4: 18 pkt.

Punktacja: 46-50 pkt - bdb(5,0); 41-45 pkt - db+(4,5); 36-40 pkt - db(4,0); 31-35 pkt - dst+(3,5); 26-30 pkt - dst(3,0); 0-25 pkt - ndst (2,0).

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`.
- Maksymalna waga archiwum 10 MB.
- Archiwum powinno być bez hasła.
- W przypadku pominięcie danego zadania, należy dodać plik o nazwie sprecyzowanej wyżej (zawartość może być pusta).
- Kod zakomentowany nie będzie sprawdzany.

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

Polecenia są na odwrocie.

Zad.1. Dane są następujące wyrazy i znaki:

`double int int suma tab a ( ) , **`

Ułóż je we właściwej kolejności (zachowując podaną krotność), aby otrzymać nagłówek funkcji `suma`, która dostaje jako argumenty kolejno tablicę tablic i liczbę całkowitą. Następnie dodaj dowolną implementację funkcji i stwórz dla niej przypadek testowy.

Zad.2. Napisz funkcję, która otrzymuje trzy argumenty:

- dwa wskaźniki na funkcje z jednym argumentem typu `double` zwracające wartość typu `double`,
- wartość `n` typu `int`,

i zwraca 1, jeżeli otrzymane w argumentach funkcje mają ten sam znak dla wartości dla liczb całkowitych od 0 do $-n$, a zwraca 0 w przeciwnym wypadku. Stwórz przypadek testowy.

Zad.3. Stwórz strukturę `Ksiazka` o dwóch polach: `tytul` (napis) i `liczba_stron` (`int`). Następnie stwórz funkcję, której argumentami jest niepusta tablica struktur `Ksiazka` oraz rozmiar tablicy. Funkcja ma zwrócić największą liczbę stron. Stwórz przypadek testowy.

Zad.4. Napisz funkcję, która otrzymuje jako argument listę z głową o elementach typu:

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

Funkcja ma usunąć z listy ostatni element nieparzysty. Stwórz przypadek testowy.