

Egzamin (pierwszy termin 2023) - Programowanie strukturalne - Zestaw E12

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 pominięcia 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:

```
* float int int const const foo a b ( ) , *
```

Ułóż je we właściwej kolejności (zachowując podaną krotkość), aby otrzymać nagłówek funkcji `foo`, która dostaje jako argumenty dwa stałe wskaźniki. Następnie dodaj dowolną implementację funkcji i stwórz dla niej przypadek testowy.

Zad.2. Napisz funkcję `reverseString`, która przyjmuje jako argument napis i zwraca nowy napis, który jest odwróconym oryginału. Oryginalny napis nie powinien ulec zmianie. Pamiętaj o prawidłowym zarządzaniu pamięcią. W zadaniu nie korzystaj z funkcji bibliotecznych związanych z napisami i znakami. Stwórz przypadek testowy dla funkcji.

Zad.3. Napisz funkcję `swapColumns`, która przyjmuje jako argumenty dwuwymiarową tablicę tablic liczb całkowitych, jej wymiary oraz dwa indeksy kolumn do zamiany miejscami. Funkcja powinna przestawić wskazane kolumny i zwrócić zmodyfikowaną tablicę. Uwzględnij sytuację, jeśli podane indeksy są nieprawidłowe - wtedy funkcja ma nic nie robić. Stwórz przypadek testowy dla funkcji.

Zad.4. Napisz funkcję, która przyjmuje jako argument listę bez głowy o elementach typu:

```
struct node {  
    float value;  
    struct node * next;  
};
```

zwraca adres ostatniej wartości ujemnej na liście. W przypadku pustej listy lub braku elementów ujemnych, funkcja ma zwrócić `NULL`. Stwórz jeden przypadek testowy.