

Egzamin (termin zerowy 2023)

- Programowanie strukturalne - Zestaw P33

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. W folderze DebugXYZ (XYZ - losowowe 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 linijek czy zaburzenie struktury kodu oznacza zero punktów za polecenie. W przypadku znaków, należy zapisać sam znak w apostrofach np. 'c' (wielkość znaków ma znaczenie).

Zad.2. Zaprojektuj funkcję `even_count`, która przyjmuje tablicę liczb całkowitych, jej rozmiar oraz wskaźnik na funkcję `int is_even(int)`. Funkcja `even_count` powinna zwracać liczbę elementów w tablicy, dla których funkcja `is_even` zwraca wartość parzystą. Stwórz przypadek testowy.

Zad.3. Zaprojektuj funkcję, której argumentem jest dwuwymiarowa kwadratowa tablica tablic (zawierająca elementy typu `int`) oraz jej wymiar $n, n > 0$. Funkcja powinna zwracać różnicę między sumą indeksów najmniejszego a sumą indeksów największego elementu w tablicy. W przypadku kilku elementów o najmniejszej lub największej wartości, powinny to być najmniejsze możliwe sumy indeksów. Przeprowadź testy na przykładowych danych.

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

```
struct node {  
    double x;  
    struct node * next;  
};
```

i zwraca najmniejszą z liczb znajdujących się na liście. W przypadku pustej listy, funkcja ma zwrócić zero. Stwórz jeden przypadek testowy.