

Egzamin (pierwszy termin 2023) - Programowanie strukturalne - Zestaw E28

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:

```
int int int void foo fun x * ( ( ( ) ) ) ,
```

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

Zad.2. Napisz bezargumentową funkcję `init_block_int`, która rezerwuje blok czterech zmiennych typu `int`. Funkcja ma ustawić kolejno w pamięci wartości 1, 2, 3 i 4. Na koniec funkcja powinna zwrócić wskaźnik na przedostatnią zmienną z bloku. Stwórz przypadek testowy w `main` tak, aby wyświetlić na konsoli wartości zmiennych przechowywanych na bloku stworzonym wewnątrz funkcji.

Zad.3. Stwórz unię `Data`, która może przechowywać wartość typu `char` lub `double`. Napisz funkcję `printData`, która jako argumenty przyjmuje unię `Data` oraz enum `DataType` określający, jaki typ danych jest przechowywany w unii. Funkcja powinna wydrukować przechowywaną wartość na konsolę. Stwórz przypadek testowy.

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

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

Funkcja ma powielić każdy ujemny element, o ile lista nie jest pusta. Stwórz przypadek testowy.

Przykład: `-1,2,-3,4 -> -1,-1,2,-3,-3,4`.