Egzamin (pierwszy termin 2023) - Programowanie strukturalne - Zestaw E13

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:

```
char * char * int int foo a b x ( , , )
```

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

Zad.2. Napisz funkcję `swap_first_last`, która przyjmuje jako argumenty trzy wskaźniki na zmienne typu `int`. Funkcja powinna zamienić miejscami wartości początkowej i ostatniej zmiennej. Stwórz przypadek testowy w `main` tak, aby wyświetlić na konsoli wartości zmiennych przed i po zamianie.

Zad.3. Stwórz strukturę `Kierowca`, która posiada pola: `imie` (tablica znaków rozmiaru 10) i `liczba_wypadkow` (typ całkowity), `liczba_przejechanych_km` (typ całkowity). Następnie napisz funkcję, której argumentami jest niepusta tablica struktur `Kierowca` oraz rozmiar tablicy. Funkcja powinna zwrócić kierowcę (jako strukturę), który przejechał najwięcej kilometrów (w przypadku kilku równych wyników, ma zwrócić wynik ostatniego). Stwórz przypadek testowy.

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

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

Funkcja ma usunąć z listy elementy parzyste. Stwórz jeden przypadek testowy.