

Egzamin (termin zerowy 2023)

- Programowanie strukturalne - Zestaw P23

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 char char * * * ( ) napis1 napis2 , foo
```

Ułóż je we właściwej kolejności, aby otrzymać nagłówek funkcji `foo`, która dostaje jako argumenty dwa napisy oraz zwraca wskaźnik na `int`. Następnie dodaj dowolną implementację funkcji i stwórz dla niej przypadek testowy.

Zad.2. Napisz funkcję w języku C, której argumentami są dwa napisy. Funkcja powinna przepisać z pierwszego napisu do drugiego napisu te znaki, które na swoich miejscach mają indeksy nieparzyste. Pamiętaj, aby drugi napis był odpowiednio duży, aby pomieścić przepisywane znaki. Stwórz przypadek testowy.

Przykład: Jeśli pierwszy napis to “abcdef”, to do drugiego napisu powinny zostać przepisane znaki “bdf”.

Zad.3. Stwórz strukturę `Telefon` z polami `marka` (tablica znaków długości 15) oraz `liczbaPołączeń` (typu `int`). Następnie, napisz dwie funkcje:

- a) `initTelefon` - funkcja, która przyjmuje dwa argumenty: markę i liczbę połączeń, i zwraca wskaźnik na nowo utworzoną strukturę `Telefon` z polami ustawionymi na wartości przekazane jako argumenty. Funkcja powinna sprawdzić, czy marka ma długość co najmniej 3 i czy liczba połączeń jest większa niż 50. Jeżeli jeden z tych warunków nie jest spełniony, funkcja powinna zwracać `NULL`.
- b) `zwiększLiczbePolaczen` - funkcja, której argumentem jest wskaźnik na strukturę `Telefon`. Funkcja powinna zwiększyć liczbę połączeń o 10.

Zad.4. Napisz funkcję, która porównuje dwie listy z głową o elementach typu:

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

i zwraca 1 jeśli listy są takiej samej długości oraz 0 w pozostałych przypadkach (także wtedy, gdy któraś z list lub obie są puste). Stwórz przypadek testowy.