

Egzamin (termin zerowy 2023) - Programowanie strukturalne - Zestaw P21

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:

```
char void int int foo a b tab a [ ] [ ] ( ) , , *
```

Ułóż je we właściwej kolejności (zachowując krotność), aby otrzymać nagłówek funkcji `foo`, która dostaje jako argumenty napis, liczbę całkowitą oraz dwuwymiarową tablicę elementów. Następnie dodaj dowolną implementację funkcji i stwórz dla niej przypadek testowy.

Zad.2. Napisz funkcję, która dostaje w argumencie napis i zamienia wszystkie występujące w nim duże litery na znak '@'. Następnie usuń wszystkie znaki '@' z napisu. W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia. Stwórz przypadek testowy

Zad.3. Zdefiniuj strukturę `Komputer` z polami `model` (tablica znaków długości 20) oraz `czasUzytkowania` (typu `int`). Następnie, stwórz dwie funkcje:

- a) `initKomputer` - funkcja, która przyjmuje dwa argumenty: `model` i `czas użytkowania`, i zwraca wskaźnik na nowo utworzoną strukturę `Komputer` z polami ustawionymi na wartości przekazane jako argumenty. Funkcja powinna sprawdzać, czy `model` ma długość co najmniej 4 i czy `czas użytkowania` jest większy niż 100. W przypadku nie spełnienia tych warunków, funkcja powinna zwracać `NULL`.
- b) `zwiększCzasUzytkowania` - funkcja, której argumentem jest wskaźnik na strukturę `Komputer`. Funkcja powinna zwiększyć `czas użytkowania` komputera o 100.

Wywołaj każdą funkcję co najmniej jeden raz.

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

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

`i` zwraca 1 jeśli ostatnie elementy na liście są równe oraz 0 w pozostałych przypadkach (także wtedy gdy któraś z list lub obie są puste). Stwórz jeden przypadek testowy.