

Egzamin (termin zerowy 2023)

- Programowanie strukturalne - Zestaw P35

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 liniijkach 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. Stwórz rekurencyjną funkcję, która przyjmuje jako argument napis. Funkcja ma zwrócić liczbę znaków będących cyframi z systemu dziesiętnego w napisie. W zadaniu nie korzystaj z funkcji bibliotecznych poza instrukcjami wejścia/wyjścia. Stwórz przypadek testowy.

Uwaga: funkcja nierekurencyjna = 0pkt.

Zad.3. Zdefiniuj strukturę `Samochod` z polami `model` (tablica znaków długości 25) oraz `przebieg` (typu `int`). Napisz funkcję `initSamochod`, która przyjmuje dwa argumenty: `model` i `przebieg`, i zwraca nowo utworzoną strukturę `Samochod` (jako wartość, nie wskaźnik) z polami ustawionymi na wartości przekazane jako argumenty. Funkcja powinna sprawdzać, czy `model` rozpoczyna się od dużej litery i czy `przebieg` jest nie większy niż 500. W przypadku nie spełnienia co najmniej jednego z tych warunków, funkcja powinna zwrócić strukturę z modelem ustawionym jako "DEFAULT" i przebiegiem równym 1000. Stwórz dwa przypadki testowe.

Zad.4. Napisz funkcję, która otrzymuje jako argument listę z głową o elementach typu:

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

Funkcja ma zdublować wartość ostatniego elementu, o ile lista jest nie pusta. W przypadku pustej listy, funkcja ma nic nie robić. Stwórz przypadek testowy.

Przykład: Dla listy 3,4,5 ma być zdublowana 5, więc po modyfikacji lista ma być postaci 3,4,5,5.