

# Egzamin (termin zerowy 2023) - Programowanie strukturalne - Zestaw P12

*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:

```
float float int () [ ] [ ] 10 , tab a fun
```

Ułóż je we właściwej kolejności, aby otrzymać nagłówek funkcji `fun`, która dostaje jako argumenty dwuwymiarową tablicę elementów wymiaru 10x10 oraz liczbę całkowitą. Następnie dodaj dowolną implementację funkcji i stwórz dla niej przypadek testowy.

Zad.2. Napisz funkcję `find_if`, która przyjmuje tablicę liczb całkowitych, jej rozmiar i wskaźnik na funkcję `int predicate(int)` jako argumenty. Funkcja `find_if` powinna zwrócić wskaźnik na pierwszy element tablicy, dla którego funkcja `predicate` zwraca wartość różną od zera. Jeżeli nie ma takiego elementu, funkcja powinna zwrócić `NULL`. Stwórz przypadek testowy.

Zad.3. Stwórz strukturę `Kierowca` o polach `imie` (napis) oraz `liczba_wypadków` (typ całkowity). Następnie stwórz funkcję, której argumentami jest niepusta tablica struktur `Kierowca` oraz rozmiar tablicy. Funkcja ma zwrócić kierowcę (jako strukturę), który miał najmniejszą liczbę wypadków (w przypadku kilku równych wyników, ma zwrócić wynik ostatniego). Stwórz przypadek testowy.

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

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

Funkcja ma zamienić wartości każdego elementu na jego wartość bezwzględną. Dla listy pustej funkcja ma nic nie robić. Stwórz przypadek testowy.