

# Egzamin (termin zerowy 2023) - Programowanie strukturalne - Zestaw P25

*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:

```
void int int int foo fun x ( ) ( ) ( ) , *
```

Ułóż je we właściwej kolejności, aby otrzymać nagłówek funkcji `fun`, której argumentami są wskaźnik na funkcję i liczba całkowita. Następnie dodaj dowolną implementację funkcji i stwórz dla niej przypadek testowy.

Zad.2. Napisz funkcję w języku C, która przyjmuje dwa napisy oraz dodatkowo liczbę całkowitą `n`. Funkcja powinna przepisać do drugiego napisu co `n`-ty znak z pierwszego napisu. Zadbaj o to, aby drugi napis miał wystarczającą ilość miejsca na przepisywanie znaków.

Przykład: Dla napisu "abcdefghijklm", liczby `n=3` oraz pustego napisu, po wykonaniu funkcji drugi napis powinien zawierać "cfil".

Zad.3. Napisz funkcję, której argumentem jest dwuwymiarowa tablica tablic (zawierająca zmienne typu `int`) oraz jej wymiary `n` i `m`. Funkcja ma zwrócić średnią elementów na głównej przekątnej. Stwórz przypadek testowy.

Zad.4. Napisz funkcję, która przyjmuje dwie listy bez głowy o elementach typu:

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

i zwraca 1 jeśli suma wszystkich elementów parzystych w obu listach jest taka sama oraz 0 w przeciwnym przypadku (także wtedy, gdy któraś z list lub obie są puste). Stwórz przypadek testowy.