

# Egzamin (pierwszy termin 2023) - Programowanie strukturalne - Zestaw E48

*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 int int suma tab a a [ ] ( ) , [ ]
```

Ułóż je we właściwej kolejności (zachowując podaną krotkość), aby otrzymać nagłówek funkcji `suma`, która dostaje jako argumenty kolejno tablicę elementów i liczbę całkowitą. Następnie dodaj dowolną implementację funkcji i stwórz dla niej przypadek testowy.

Zad.2. Napisz funkcję `copy_squares()`, która przyjmuje jako argumenty tablicę wartości całkowitych oraz jej rozmiar. Funkcja powinna utworzyć tablicę dynamiczną, gdzie każdy element nowej tablicy będzie zawierał kwadrat wartości z odpowiadającej pozycji w wejściowej tablicy. Następnie funkcja powinna zwrócić wskaźnik na pierwszy element nowej tablicy. Stwórz przypadek testowy.

Zad.3. Utwórz strukturę `Temperature` zawierającą pola `city` (typu `char*`) i `temperature` (typu `float`). Napisz funkcję `max_temperature()`, która przyjmuje jako argument niepustą tablicę struktur `Temperature` i jej rozmiar, a następnie oblicza i zwraca maksymalną temperaturę spośród wszystkich miast. Stwórz przypadek testowy, w którym utworzysz tablicę struktur `Temperature` z różnymi miastami i temperaturami, wywołasz funkcję `max_temperature()` i wyświetlisz na konsoli maksymalną temperaturę.

Zad.4. Napisz funkcję, która otrzymuje jako argument listę bez głowy o elementach typu:

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

oraz wartość całkowitą `value`. Funkcja powinna wyświetlić na konsoli adresy elementów, których wartość jest większa niż podana wartość `value`. Stwórz przypadek testowy.