

# Egzamin (drugi termin 2023)

## - Programowanie strukturalne - Zestaw M12

*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.

1. W folderze DebugXYZ (XYZ - losowe znaki) znajduje się projekt z kodem w języku C. W pliku main.c w niektórych liniach są komentarze. Twoim zadaniem jest wpisanie wartości odpowiednich zmiennych po wykonaniu konkretnej linii kodu. Dopisanie nowych linii czy zaburzenie struktury kodu oznacza zero punktów za polecenie. W przypadku znaków, należy zapisać sam znak w apostrofach np. 'c' (wielkość znaków ma znaczenie).
2. Napisz funkcję, której argumentami są dwa napisy. Funkcja ma przepisać z pierwszego napisu znaki na parzystych indeksach do drugiego napisu. Stwórz przypadek testowy.

Przykład: Pierwszy napis to "informatyka". Drugi napis po przepisaniu ma wyglądać następująco: "ifraya".

3. Stwórz strukturę **Budynek** o dwóch polach: **nazwa** (napis) oraz **liczba\_pieter** (dowolny typ całkowity). Następnie stwórz funkcję, której argumentami są niepusta tablica struktur **Budynek** oraz jej rozmiar. Funkcja ma zwrócić indeks elementu tablicy o najmniejszej liczbie pieter (możesz założyć, że taki element jest tylko jeden). Stwórz przypadek testowy.
4. Napisz funkcję, która przyjmuje jako argument listę z głową o elementach typu:

```
struct listnode {  
    int y;  
    struct listnode * next;  
};
```

oraz dwie liczby całkowite **a** i **b**. Funkcja ma dodać na początek listy dwa nowe elementy i ich wartości ustawić odpowiednio z podanych argumentów. Stwórz jeden przypadek testowy.