

# Egzamin (przykładowy) - Programowanie strukturalne - Zestaw 1

*Zadanie 1: 10 pkt. Zadanie 2: 12 pkt. Zadanie 3: 13 pkt. Zadanie 4: 15 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).*

Zad.1. Dane są następujące wyrazy i znaki:

```
char char char * * * ( ) napis1 napis2 , foo
```

Ułóż je we właściwej kolejności, aby otrzymać nagłówek funkcji foo, która dostaje jako argumenty dwa napisy oraz zwraca wskaźnik na `char`. Następnie dodaj dowolną implementację funkcji i stwórz dla niej przypadek testowy.

Zad.2. Napisz funkcję, która jako argument otrzymuje dodatnią liczbę całkowitą  $n$  i zwraca liczbę równą sumie sześciąt kolejnych liczb naturalnych dodatnich mniejszych lub równych  $n$  ( $1^3 + 2^3 + \dots + n^3$ ). Nie korzystaj z żadnych gotowych funkcji bibliotecznych ani wbudowanych poza instrukcjami wejścia/wyjścia. Stwórz dwa przypadki testowe dla funkcji.

Zad.3. Napisz funkcję, której argumentem jest dwuwymiarowa tablica tablic typu `int` i jej wymiary  $n$  i  $m$ . Funkcja ma zwrócić różnicę między największym a najmniejszym elementem tablicy. Stwórz przypadek testowy.

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

```
struct element {  
    float x;  
    struct element * next;  
};
```

Funkcja ma wyświetlić adres pierwszego i ostatniego elementu. W przypadku pustej listy, ma być wyświetlone odpowiednio NULL. Stwórz przypadek testowy.