

Egzamin (pierwszy termin) - Programowanie Strukturalne - Zestaw G28

Zadanie 1 i 2: po 10 pkt. Zadanie 3 i 4: po 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:

```
int int int int int fun f2 f1 n m k ( ) ( ) ( ) ( ) ( ) , , * *
```

Ułóż je we właściwej kolejności, aby otrzymać nagłówek funkcji fun, która dostaje jako argumenty dwa wskaźniki na funkcję i liczbę całkowitą i zwraca liczbę całkowitą. 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 sześcianów kolejnych liczb naturalnych dodatnich mniejszych lub równych n ($1^3 + 2^3 + \dots + n^3$). Nie korzystaj z żadnych gotowych funkcji bibliotecznych ani wbudowanych. Stwórz dwa przypadki testowe dla funkcji.

Przypadki testowe:

n	wyjście
1	1
2	9
3	36

Zad.3. Napisz funkcję, której argumentem jest dwuwymiarowa tablica elementów typu `int` i jej wymiary n i m . Funkcja ma zwrócić wskaźnik na element w ostatnim wierszu i pierwszej kolumnie (kolumnie o indeksie 0). Stwórz przypadek testowy prezentujący wartość przechowywaną na adresie zwróconym przez funkcję pomniejszonym o 1 (adres zwrócony przez funkcję należy najpierw dekrementować, a potem wyświetlić wartość z niego).

Zad.4. Napisz funkcję, która porównuje dwie listy z głową o elementach typu:

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

i zwraca wskaźnik na nową listę z głową zawierającą maksimum odpowiadających sobie elementów (możesz przyjąć, że obie listy mają tyle samo elementów). Stwórz jeden przypadek testowy.