

Egzamin (pierwszy termin) - Programowanie Strukturalne - Zestaw G09

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. W folderze PoprawG09 znajduje się kod w języku C, który nie spełnia zasad kompilacji. Popraw kod modyfikując dokładnie 1 linijkę tak, aby się kompilował. Zabronioną operacją jest komentowanie kodu. Do zmodyfikowanych linii zaliczają się zarówno linie istotne ze względu na kompilację jak i te nieistotne (np. dodanie spacji przed operatorem może być operacją nieistotną ze względu na kompilację, ale będzie liczone jako zmodyfikowana linijka).

Zad.2. Napisz funkcję rekurencyjną, która dla otrzymanej w argumencie nieujemnej liczby całkowitej n zwraca wartość elementu o indeksie n ciągu zdefiniowanego w następujący sposób

$$a_0 = 1, a_1 = -1$$

$$a_n = \begin{cases} a_{n-2} + n, & n - \text{parzyste} \\ a_{n-2} * n, & n - \text{nieparzyste} \end{cases}$$

Stwórz dwa przypadki testowe dla funkcji.

Zad.3. Napisz funkcję, której argumentem jest dwuwymiarowa tablica tablic i jej wymiary n i m . Funkcja ma zwrócić sumę elementów znajdujących się na brzegach tablicy i miejscach o indeksach nieparzystych. Stwórz przypadek testowy.

Przykład: dla poniższej tablicy ma wyjść: $3 + 8 + (-1) + 3 + 5 + (-9) = \dots$

2	3	-2	8
-1	8	-4	3
3	5	-2	-9

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

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

Funkcja ma usunąć z listy pierwszy element nieparzysty. Stwórz przypadek testowy.