

Wstęp do programowania w języku C

Grupa MSz w poniedziałki

Lista 6 na zajęcia 25.11.2019

Zadanie 1. (10 punktów na pierwszej pracowni, 5 punktów na drugiej)

Na wejściu dana jest liczba $n \leq 20$ oraz n liczb całkowitych. Napisz program, który dla podanego ciągu znajdzie takie wstawienie operatorów arytmetycznych (+, -, *, /) między liczbami by wynik wyrażenia wynosił 1. Ponadto wstawienie musi spełniać warunek, że żaden operator nie występuje dwa razy bezpośrednio po sobie.

Jako wynik powinien wypisać otrzymane wyrażenie lub stosowną informację, jeśli nie da się uzyskać jedynki. Proszę zwrócić uwagę, że może wystąpić dzielenie przez 0; wówczas wyrażenie na pewno nie daje jedynki.

Dla uproszczenia przyjmujemy, że wszystkie operacje są wykonywane od lewej do prawej (mają ten sam priorytet). Obliczenia wykonujemy na typie `int` nie przejmując się możliwym przepełnieniem; dzielenie jest więc całkowitoliczbowe.

Przykład 1:

3
2 2 1

Wynik:

2 - 2 + 1

Przykład 2:

5
10 8 0 2 1

Jeden z możliwych wyników:

10 + 8 * 0 + 2 - 1

Test 3:

16
9 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Test 4.

20

99 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Zadanie 2. (10 punktów)

Dane są liczby m (szerokość) i n (wysokość) oraz tablica liczb o tych wymiarach. Należy zastąpić wszystkie zera w tablicy niezerowymi liczbami ze zbioru $\{-9, -8, \dots, -1, 1, 2, \dots, 9\}$ tak żeby suma liczb w każdym wierszu i w każdej kolumnie równała się zero. Wynikiem jest tabelka z dowolnym rozwiązaniem, lub stosowny komunikat jeśli rozwiązanie nie istnieje.

Tym razem nie ma podanych ograniczeń na wielkość danych. Program powinien być rozsądny i działać dla małych danych; w szczególności powinien rozwiązywać podane przykłady i testy. Można zastosować rekurencję z szybkim wycofywaniem gdy wybrana liczba na pewno nie daje zerowej sumy.

Przykład 1:

```
3 3
0 1 2
4 0 -1
0 0 0
```

Wynik:

```
-3 1 2
4 -3 -1
-1 2 -1
```

Przykład 2:

```
4 4
0 1 1 1
0 0 1 1
0 0 0 1
0 0 0 0
```

Wynik (jeden z wielu):

```
-3 1 1 1
-1 -1 1 1
1 -1 -1 1
3 1 -1 -3
```

Przykład 3:

```
5 4
0 0 0 0 1
```

```

1 1 1 1 0
2 2 2 2 0
3 3 3 3 0

```

Wynik: brak rozwiązania.

Test wydajnościowy 4:

```

10 10
0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 0

```

Test wydajnościowy 5:

```

5 10
0 0 0 0 0
0 0 0 0 0
1 1 1 1 0
1 1 1 1 0
1 1 1 1 0
1 1 1 1 0
1 1 1 1 0
1 1 1 1 0
1 1 1 1 0
1 1 1 1 0
1 1 1 1 0

```