

Подготовка за контролна работа

Задача 1.

Дадена е структурата

```
struct Gas
{
    int counterId;
    double quantity;
};
```

която описва единична продажба на гориво в бензиностанцията "Газ при нас".

Полето `counterId` (с валидни стойности ≥ 1) означава номер на колонка, а `quantity` – количество заредено гориво в литри.

В текстовия файл `sales.txt` са записани последователно редове от вида

`<counterId> <quantity>`

описващи информация за продажбите в "Газ при нас" в даден ден.

Да се напишат следните функции:

а) **`double totalForCounter(int counterId, double pricePerLiter)`**, която намира реализирания приход от колонка с даден номер, при дадена цена на литър гориво;

б) **`void writeAverages()`**, която записва в двоичния файл `averages.bin` числа от тип `double`, като $(i-1)$ -вото поред число означава средното количество гориво, заредено на колонка с номер i , според текстовия файл `sales.txt`. В случай, че във файла `sales.txt` няма информация за колонка с номер i , на съответната позиция в `averages.bin` да се запише числото 0.

Задача 2.

Дадени са два стека от числа, подредени в намаляващ ред от върха към дъното. Да се напише функция **`mergeStacks`**, която чрез операциите `push` и `pop` построява нов стек, който се състои от всички елементи на дадените два стека, подредени в нарастващ ред от върха към дъното.

Пример: 5, 3, 1 6, 2, 1 -> 1, 1, 2, 3, 5, 6

Задача 3.

Дадени са бинарните операции:

$$a\$b = \min(a, b) \text{ и } a@b = (a+b)\%10,$$

където a и b са едноцифрени числа.

Да се напише програма, която изчислява изрази от вида:

`<израз> = <цифра> | (<израз><операция><израз>)`

`<операция> = $ | @`

Пример: 5; (2\$3); (2@(3\$4)) и т.н.

Задача 4.

Да се напише програма, която при получен "компресиран" низ на стандартния вход, извежда на стандартния изход декомпресираната му версия.

Компресираният низ съдържа 2 вида конструкции:

- букви, които са символ от 'A' до 'Z' и се декомпресират до същата буква;
- групи, които започват с **число**, последвано от компресиран низ в скоби. Декомпресират се като се декомпресира низа в скобите и се повтори толкова пъти, колкото е числото.

Пример:

A -> A

AB -> AB

3 (A) -> AAA

3 (AB2 (B)) -> ABVBABVBABVB

12 (ZQ) -> ZQZQZQZQZQZQZQZQZQZQZQ

Задача 5.

Дадена е редицата числа, чиито членове се получават по-следния начин:

- първият елемент е N;
- вторият се получава като съберем N с 1;
- третият – като се умножи първия с 2 и така последователно всеки елемент се събира с 1 и се добавя в края на редица, след което се умножава по 2 и отново се добавя в редицата.

Да се напише програма, която за дадено N и r намира r-тия пореден елемент на редицата.

Задача 6.

Банка разполага с определен брой гишета за работа с клиенти (може да се подава като аргумент на класа Банка, ако има такъв). На всеки 5 мин. в банката пристига нов клиент. Клиентите се подреждат в опашка, обслужват се по реда на тяхното пристигане. Всеки клиент пристига в определено време и се нуждае от определено време за обслужване, което се генерира случайно и е в интервала между 10 и 20 мин. Всяка минута се проверява дали има свободно гише, ако има такова, то поема следващия клиент. Ако няма свободно гише, за всяко гише се преизчислява оставащото време за обслужване на текущия клиент. Ако времето за обслужване на клиент на дадено гише е приключило, то гишето ще бъде свободно на следващата итерация.

Една итерация на цикъла е с продължителност 1 мин.

Да се напише програма, която симулира работата на банката за период от 1 час (или повече) и изчислява средното време за чакане на всеки клиент.