10836. Лето

Бруно и его друзья играют с водяными пистолетами. Они заядлые геймеры, поэтому их игра вовсе не обычная водяная битва, а почти настоящая видеоигра. Ребята даже пригласили модератора, чтобы следить за ходом сражения.

В начале игры участники делятся на две команды – “ананас” и “черника”.

Во время игры модератор фиксирует моменты времени, когда один игрок делает выстрел в другого. Как и в видеоиграх, за успешные выстрелы начисляются очки.

Если игрок одной команды попадает в игрока противоположной команды, его команда получает 100 очков. Однако, если тот же игрок в течение следующих 10 секунд снова попадёт в кого-то из соперников, этот выстрел считается двойным, и команда получает дополнительные 50 очков. Игрок может совершать несколько двойных выстрелов подряд – за каждый из них его команда получает ещё по 50 очков сверх обычных 100.

**Вход.** Первая строка содержит одно целое число *n* (1 ≤ *n* ≤ 100) – количество выстрелов, сделанных во время игры.

Каждая из следующих  строк содержит три целых числа *ti*, *ai*, *bi* (0 ≤ *ti* ​≤ 1000, 1 ≤ *ai*, *bi* ≤ 8) обозначающих, что игрок с номером *ai* выстрелил в игрока с номером *bi* в момент времени *ti* (в секундах).

Игроки команды “ананас” имеют номера от 1 до 4, а игроки команды “черника” – от 5 до 8. Гарантируется, что в каждом выстреле игроки *ai* и *bi* принадлежат разным командам.

Числа  различны и заданы в порядке возрастания.

**Выход.** Выведите два целых числа – общий счёт команды “ананас” и общий счёт команды “черника”.

|  |  |
| --- | --- |
| **Пример входа 1** | **Пример выхода 1** |
| 310 1 620 1 721 8 1 | 250 100 |
|  |  |
| **Пример входа 2** | **Пример выхода 2** |
| 310 2 515 2 625 2 5 | 400 0 |
|  |  |
| **Пример входа 3** | **Пример выхода 3** |
| 210 5 211 6 3 | 0 200 |

## РЕШЕНИЕ

**математика**

# Анализ алгоритма

Для каждого из 8 игроков будем хранить информацию о времени его последнего выстрела. Например, в массиве prevTime[*i*] будем записывать момент времени (в секундах), когда игрок *i* сделал свой последний выстрел.

Далее последовательно обрабатываем все *n* выстрелов. Пусть текущий выстрел представлен тройкой (*t*, *a*, *b*). Тогда:

* при выстреле команде игрока *a* начисляется 100 очков;
* если игрок *a* совершает двойной выстрел (выполняется условие *t* – prevTime[*a*] ≤ 10), его команде дополнительно начисляется 50 очков;
* после этого обновляем значение prevTime[*a*] = *t*, так как игрок *a* сделал свой последний выстрел в момент времени *t*.

**Пример**

В первом примере на 10-й и 20-й секундах игрок 1 стреляет в игроков 6 и 7 из противоположной команды. За каждый выстрел команда “Ананас” получает по 100 очков. Поскольку оба выстрела были сделаны с разницей не более 10 секунд, команда получает дополнительные 50 очков. Итого: 250 = 2 × 100 + 50. Команда “Черника” совершила лишь один выстрел по сопернику и набрала 100 очков.

Во втором примере игрок 2 сделал два двойных выстрела подряд, поэтому команда “Ананас” заработала в сумме 3 × 100 + 2 × 50 = 400 очков.

**Реализация алгоритма**

В prevTime[*i*] будем хранить время (в секундах), когда игрок *i* совершил свой последний выстрел.

int prevTime[10];

Читаем количество выстрелов *n*.

scanf("%d", &n);

Для каждого игрока начальное значение времени последнего выстрела устанавливаем равным -∞.

for (i = 1; i < 10; i++)

 prevTime[i] = -10000;

Очки команд “Ананас” и “Черника” будем хранить в переменных *pa* и *pb* соответственно.

pa = pb = 0;

Обрабатываем все *n* выстрелов.

while (n--)

{

Игрок *a* совершает выстрел в игрока *b* в момент времени *t*.

 scanf("%d %d %d", &t, &a, &b);

Команде, к которой принадлежит игрок *a*, начисляется 100 очков.

 if (a < b) pa += 100;

 else pb += 100;

Если игрок *a* совершает выстрел в течение 10 секунд после своего предыдущего, его команда получает дополнительные 50 очков.

 if (t - prevTime[a] <= 10)

 if (a < b) pa += 50;

 else pb += 50;

После этого обновляем информацию о том, что игрок *a* сделал выстрел в момент времени *t*.

 prevTime[a] = t;

}

Выводим ответ.

printf("%d %d\n", pa, pb);