10836. Summer

Bruno and his friends are playing with water guns. They are avid gamers, so their game is not an ordinary water fight but almost like a real video game. They even invited a moderator to keep track of the gameplay.

At the beginning of the game, the players are divided into two teams – “Pineapple” and “Blueberry”.

During the game, the moderator records the moments when one player shoots another. As in video games, successful shots earn points.

If a player from one team hits a player from the opposite team, their team earns 100 points. However, if the same player hits another opponent within the next 10 seconds, that shot is considered a double shot, and the team receives an additional 50 points. A player can make several double shots in a row – for each of them, their team gains another 50 points in addition to the regular 100.

**Input.** The first line contains one integer *n* (1 ≤ *n* ≤ 100) – the number of shots made during the game.

Each of the next  lines contains three integers *ti*, *ai*, *bi* (0 ≤ *ti* ​≤ 1000, 1 ≤ *ai*, *bi* ≤ 8), meaning that player *ai* shot player *bi* at time *ti* (in seconds).

Players from team “Pineapple” are numbered from 1 to 4, and players from team “Blueberry” are numbered from 5 to 8. It is guaranteed that in each shot, players *ai* and *bi* belong to different teams.

All values of  are distinct and given in increasing order.

**Output.** Print two integers – the total score of team “Pineapple” and the total score of team “Blueberry”.

|  |  |
| --- | --- |
| **Sample input 1** | **Sample output 1** |
| 310 1 620 1 721 8 1 | 250 100 |
|  |  |
| **Sample input 2** | **Sample output 2** |
| 310 2 515 2 625 2 5 | 400 0 |
|  |  |
| **Sample input 3** | **Sample output 3** |
| 210 5 211 6 3 | 0 200 |

## SOLUTION

**mathematics**

# Algorithm analysis

For each of the 8 players, we will store information about the time of their last shot. For example, in the array prevTime[*i*], we will record the moment in time (in seconds) when player *i* made their last shot.

Next, we process all *n* shots sequentially. Let the current shot be represented by a triple (*t*, *a*, *b*). Then:

* when player *a* makes a shot, their team receives 100 points;
* if player *a* performs a double shot (that is, if the condition *t* – prevTime[*a*] ≤ 10 holds), their team receives an additional 50 points;
* after that, we update the value of prevTime[*a*] = *t*, since player *a* made their most recent shot at time *t*.

**Example**

In the first example, at the 10th and 20th seconds, player 1 shoots players 6 and 7 from the opposing team. For each shot, team “Pineapple” earns 100 points. Since both shots were made within a 10-second interval, the team receives an additional 50 points. Total: 250 = 2 × 100 + 50. Team “Blueberry” made only one shot at an opponent and scored 100 points.

In the second example, player 2 made two consecutive double shots, so team “Pineapple” earned a total of 3 × 100 + 2 × 50 = 400 points.

**Algorithm implementation**

In prevTime[*i*], we’ll store the time (in seconds) when player *i* made their last shot.

int prevTime[10];

Read the number of shots *n*.

scanf("%d", &n);

For each player, the initial value of the last shot time is set to -∞.

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

 prevTime[i] = -10000;

Store the scores of teams “Pineapple” and “Blueberry” in the variables *pa* and *pb*, respectively.

pa = pb = 0;

Process all *n* shots.

while (n--)

{

Player *a* makes a shot at player *b* at time *t*.

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

The team to which player *a* belongs receives 100 points.

 if (a < b) pa += 100;

 else pb += 100;

If player *a* makes a shot within 10 seconds after their previous one, their team earns an additional 50 points.

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

 if (a < b) pa += 50;

 else pb += 50;

After that, update the information that player *a* made a shot at time *t*.

 prevTime[a] = t;

}

Print the answer.

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