**算法分析与设计实验报告**

**第 4 次实验**

**10526** The Uxuhul Voting System

|  |
| --- |
| **Problem description** |
| One of the world's first civilisations was that of the ancient Uxuhul indians, in the jungles of Central America. The Uxuhul culture flourished for almost a thousand years, with its golden era around 3200 BC. Each year high priests, representing different parts of the realm, gathered in the capital to vote on important matters. There were always exactly three issues to vote on, and each issue was a simple yes/no question. The three issues were all decided in one single voting round, in the following manner:  All the priests would gather in a large room, with a table in the centre. On the table lie three flat stones, each with one white, and one black side. Each stone represented an issue, with the black side up symbolising the outcome 'no' and the white side the outcome 'yes'. Initially all stones had the black side up, representing negative outcome on all issues. In order of age, from youngest to oldest, each priest voted, by turning exactly one stone, thus (temporarily) changing the outcome of one issue. It was not allowed to skip the vote, each priest had to flip a stone. The colours of the side facing up on the stones after the oldest priest had flipped the stone of his choice, determined the final outcome of the voting on the three issues.  The Uxuhul political life was rather complex, with a lot of lobbying by different interest groups. Lobbying (and bribery), gave each priest preferences between the eight different outcomes of the three issues. Religious rules forced each priest to publicly state his preference order between the eight outcomes, before the voting process started. Based on knowledge of each other priest's preferences, it was possible for the priests to choose a stone to flip in the way that it gained their interest best. As the Uxuhul were advanced logicians, with deep knowledge of game theory, each priest was actually able to vote in the optimal way!  Eventually, the complexity and rigidity of their political system brought the Uxuhul culture to collapse. Today only ruins remain of their cities and temples. Historians and archaeologists try to understand more about their history by studying the results from their yearly votes. However, only records of the priests' preferences remain, not the actual outcomes of the votes. Now it is your task to find a way to recover the outcomes. |
| **Input** |
| The input begins with an integer n, 100>n>1, the number of voting rounds. Starting on the next line, n problems follow, one by one. Each voting starts with a line containing an integer m, 100>m>1, stating the number of voting priests. Then there are m lines, defining the preference order for the outcomes of the three issues for each voter, in order, starting with the voter who will begin the voting procedure. The preferences for each outcome (NNN, NNY, NYN, NYY, YNN, YNY, YYN, and YYY respectively, where ‘N’ stands for no and ‘Y’ for yes), are given as an integer in the range [1…8]. Lower numbers indicate higher preference. The preferences for the eight outcomes are given in the same order as the enumeration of the outcomes above. |
| **Output** |
| For each of the n voting problems, output one line describing the outcome of the votes for the three issues, 'Y' for yes, and 'N' for no. |
| **Sample Input** |
| 2  4  8 7 6 5 4 3 2 1  8 6 3 1 2 4 5 7  8 3 6 5 1 2 7 4  1 2 3 4 5 6 7 8  1  1 2 3 4 5 6 7 8 |
| **Sample Output** |
| NYY  NNY |

**附录：完整代码**

*//The Uxuhul Voting System*

#include <cstdlib>

#include <cstdio>

using namespace std;

int arr[105][8];

char ch[8][4] = {"NNN", "NNY", "NYN", "NYY", "YNN", "YNY", "YYN", "YYY"};

int re[8], cmp[8], t[4];

int main()

{

    int ca, n;

    while (scanf("%d", &ca) != EOF)

    {

        while (ca--)

        {

            for (int i = 0; i < 8; i++)

                re[i] = i;

            scanf("%d", &n);

            for (int i = 0; i < n; i++)

                for (int j = 0; j < 8; j++)

                    scanf("%d", &arr[i][j]);

            for (int i = n - 1; i >= 0; i--)

            {

                for (int j = 0; j < 8; j++)

                {

                    for (int k = 0; k < 3; k++)

                    {

                        t[k] = re[1 << k ^ j];

                    }

                    cmp[j] = t[0];

                    if (arr[i][cmp[j]] > arr[i][t[1]])

                        cmp[j] = t[1];

                    if (arr[i][cmp[j]] > arr[i][t[2]])

                        cmp[j] = t[2];

                }

                for (int j = 0; j < 8; j++)

                    re[j] = cmp[j];

            }

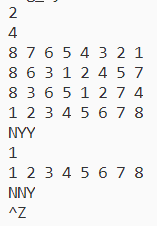
            printf("%s\n", ch[re[0]]);

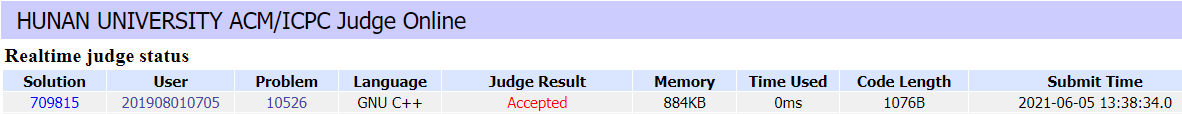
        }

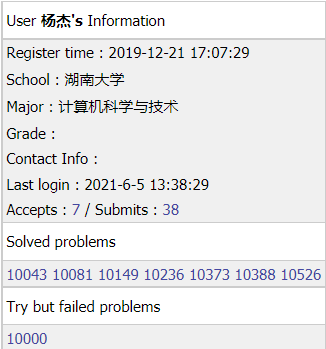
    }

    return 0;

}



****

****