# Завдання IІ етапу Всеукраїнської учнівської олімпіади

# з інформатики 2017/2018 н.р.

<http://134.249.159.199/cgi-bin/new-client?contest_id=56>

|  |  |
| --- | --- |
| Логін | school2018-2019-1 school2018-2019-2…school2018-2019-50 |
| Пароль | 1 |

**Задача A. Покер (100 балів)**

**Ім’я вхідного файлу: іnput.txt**

**Ім’я вхідного файлу: output.txt**

**Ліміт часу: 1с.**

Задано 5 цілих чисел. Серед них:

* якщо однакові 5, то вивести "Impossible", інакше
* якщо однакові 4, то вивести "Four of a Kind", інакше
* якщо однакові 3 и 2, то вивести "Full House", інакше
* якщо є 5 послідовних, то вивести "Straight", інакше
* якщо однакові 3, то вивести "Three of a Kind", інакше
* якщо однакові 2 і 2, то вивести "Two Pairs", інакше
* якщо однакові 2, то вивести "One Pair", інакше
* вивести "Nothing".

**Вхідні дані**

У першому рядку задано 5 чисел (від 1 до 13 включно) через пропуск.

**Вихідні дані**

Виводиться один рядок - результат аналізу.

**Приклади**

|  |  |
| --- | --- |
| іnput.txt | output.txt |
| 1 3 9 3 2 | One Pair |
| 1 5 5 4 4 | Two Pairs |
| 10 11 12 13 1 | Nothing |

|  |  |
| --- | --- |
| **C++** | **Python** |
| #include <iostream>using namespace std;int a[14];int main(){ int t,k2=0,k3=0,k4=0,k5=0,k0=0; for(int i=1;i<=5;i++) {cin>>t;a[t]++;} for(int i=1;i<=13;i++) { if(a[i]==2)k2++; if(a[i]==3)k3++; if(a[i]==4)k4++; if(a[i]==5)k5++; } int i=1; while(a[i]==0)i++; if(i<=9 && a[i]==1 && a[i+1]==1 && a[i+2]==1 && a[i+3]==1 && a[i+4]==1) k0=1;if(k5==1) cout<<"Impossible"; elseif(k4==1) cout<< "Four of a Kind"; elseif(k3==1 && k2==1) cout<< "Full House"; elseif(k3==1) cout<< "Three of a Kind"; elseif(k2==2) cout<< "Two Pairs"; elseif(k2==1) cout<< "One Pair"; elseif(k0==1) cout<< "Straight"; elsecout<< "Nothing"; return 0;} | import logginglogging.basicConfig(level=logging.DEBUG)inF = open('input.txt', 'r')inC = inF.read()inC = inC.replace('\n', '')inC = list(inC.split(' '))inC.sort()logging.debug(inC)if inC[0] == inC[1] == inC[2] == inC[3] == inC[4]:    outF = open('output.txt', 'w')    outF.write('Impossible')elif inC[0] == inC[1] or inC[1] == inC[2] or inC[2] == inC[3] or inC[3] == inC[4]:    if inC[0] != inC[1] or inC[2] != inC[3] or inC[4] != inC[0]:        outF = open('output.txt', 'w')        outF.write('One Pair')    if inC[0] == inC[1] == inC[2] == inC[3] and inC[0] != inC[4]:        outF = open('output.txt', 'w')        outF.write('Four of a Kind')    elif inC[1] == inC[2] == inC[3] == inC[4] and inC[0] != inC[4]:        outF = open('output.txt', 'w')        outF.write('Four of a Kind')    elif inC[1] == inC[2] == inC[3] and inC[0] != inC[3] and inC[4] != inC[3] and inC[0] != inC[4]:        outF = open('output.txt', 'w')        outF.write('Three of a Kind')    elif inC[0] == inC[1] == inC[2] and inC[3] != inC[0] and inC[4] != inC[0] and inC[3] != inC[4]:        outF = open('output.txt', 'w')        outF.write('Three of a Kind')    elif inC[2] == inC[3] == inC[4] and inC[2] != inC[0] and inC[2] != inC[1] and inC[0] != inC[1]:        outF = open('output.txt', 'w')        outF.write('Three of a Kind')    elif inC[0] == inC[1] and inC[2] == inC[3] and inC[4] != inC[0] != inC[2]:        outF = open('output.txt', 'w')        outF.write('Two Pairs')    elif inC[0] == inC[1] and inC[3] == inC[4] and inC[2] != inC[0] != inC[3]:        outF = open('output.txt', 'w')        outF.write('Two Pairs')    elif inC[1] == inC[2] and inC[3] == inC[4] and inC[0] != inC[1] != inC[3]:        outF = open('output.txt', 'w')        outF.write('Two Pairs')    elif inC[0] == inC[1] and inC[2] == inC[3] == inC[4] and inC[0] != inC[2]:        outF = open('output.txt', 'w')        outF.write('Full House')    elif inC[3] == inC[4] and inC[0] == inC[1] == inC[2] and inC[0] != inC[3]:        outF = open('output.txt', 'w')        outF.write('Full House')elif inC[0] != inC[1] and inC[1] != inC[2] and inC[2] != inC[3] and inC[3] != inC[4]:    if inC[1] != int(inC[0])+1 and inC[2] != int(inC[1])+1 and inC[3] != int(inC[2])+1 and inC[4] != int(inC[3])+1:        outF = open('output.txt', 'w')        outF.write('Nothing')    if int(inC[1]) == int(inC[0])+1 and int(inC[2]) == int(inC[1])+1 and int(inC[3]) == int(inC[2])+1 and int(inC[4]) == int(inC[3])+1:        outF = open('output.txt', 'w')        outF.write('Straight')    if int(inC[0]) + int(inC[1]) + int(inC[2]) + int(inC[3]) + int(inC[4]) >= 40:        outF = open('output.txt', 'w')        outF.write('Straight') |

**Задача B. Рада школи (100 балів)**

**Ім’я вхідного файлу: іnput.txt**

**Ім’я вхідного файлу: output.txt**

**Ліміт часу: 1 с.**

До керівництва ради школи входять батьки, учні та вчителі, причому батьків повинно бути не менше 1/3 від загальної кількості членів ради. На поточний момент часу до ради школи входить N людей, з них K батьків. Визначте, скільки батьків потрібно додатково ввести в число ради школи, щоб їх кількість була не меншою 1/3 від загального числа всіх учасників.

**Вхідні дані**

Вхідні дані містять два числа N і K (N › 0, 0 ≤ K ≤ N≤100000) записані через пропуск.

**Вихідні дані**

Вихідні дані містять одне число - мінімальна кількість батьків, яких потрібно долучити до числа ради школи.

**Приклад**

|  |  |
| --- | --- |
| іnput.txt | output.txt |
| 27 7 | 3 |

|  |  |
| --- | --- |
| **C++** | **Python** |
| #include <iostream>using namespace std;int main(){ long long n,k,x; cin>>n>>k; x=n/3-k; if(x<0)x=0; k=k+x; n=n+x; cout<<x<<" "<<n<<" "<<k<<endl; while(n>k\*3) { n++; k++; x++; cout<<x<<" "<<n<<" "<<k<<endl; } cout << x << endl; return 0;} |

|  |  |
| --- | --- |
|  | inp = open("input.txt",'r')out = open("output.txt",'w')dani = inp.read().split()r = (int(dani[0])-int(dani[1]))x = int((r\*3)/2)if x\*2 != r\*3 :        x+=1x -= int(dani[0])if x < 0:        x=0out.write(str(x)) |

 |

**Задача С. N-та цифра (100 балів)**

**Ім’я вхідного файлу: іnput.txt**

**Ім’я вхідного файлу: output.txt**

**Ліміт часу: 1с.**

Дано початковий рядок 1, над яким застосовують наступну операцію: приписують праворуч такий же рядок, але з заміною нулів на одиниці, а одиниць на нулі. Цю операцію повторюють нескінченну кількість разів. Ваше завдання полягає в знаходженні N-тої цифри в отриманому рядку.

**Вхідні дані**

В першому рядку записано кількість тестів K. В наступних K рядках записано число N, номер цифри в K-му рядку (1≤K≤100, 1≤N≤1000000000).

**Вихідні дані**

В результаті вивести K рядків, в кожному з яких N-та цифра, відповідного рядка.

**Приклади**

|  |  |
| --- | --- |
| іnput.txt | output.txt |
| 17 | 1 |
| 212 | 10 |

//#include <iostream>

#include <fstream>

#include <string.h>

using namespace std;

ifstream cin("input.txt");

ofstream cout("output.txt");

int main()

{

char \*a=new char[100000001];

int k;

cin>>k;

for(int j=0;j<k;j++){

long long n;

cin>>n;

long long int l=1;

a[0]='1';

while(l<n)

{

//l=strlen(a);

for(long long int i=0;i<l;i++){

 a[i+l]=(49-int(a[i]))+48;

 //cout<<49-int(a[i])<<"\*\*"<<endl;

}

l=l\*2;

//for(int i=0;i<l;i++)cout<<a[i];cout<<"---"<<endl;

}

cout<<a[n-1]<<endl;

}

 return 0;

}

**Задача D. Порожній прямокутник (100 балів)**

**Ім’я вхідного файлу: іnput.txt**

**Ім’я вхідного файлу: output.txt**

**Ліміт часу: 2с.**

 Задано N різних точок на площині (2≤N≤1000). Виберемо будь-які дві точок з даної множини з координатами (x1, y1) і (x2, y2). Для цієї пари точок можна побудувати прямокутник зі сторонами, паралельними осям координат, так що обрані точки будуть перебувати в протилежних кутах прямокутника. Прямокутник, побудований таким чином, назвемо порожнім, якщо всередині нього і на його границі немає інших заданих точок. Вам потрібно визначити, скільки різних порожніх прямокутників можна побудувати із заданого множини точок.

**Вхідні дані**

 Вхідні дані складаються з декількох рядків. Перший рядок файлу містить величину N(кількість точок). У кожному наступному рядкові через пропуск задаються координати X і Y однієї точки з множини цілих чисел, що не перевищують по модулю 1000000 (в 50% тестів координати не перевищують по модулю 10000).

**Вихідні дані**

Вихідні дані містять єдиний рядок зі знайденим числом порожніх прямокутників

**Приклад**

|  |  |
| --- | --- |
| іnput.txt | output.txt |
| 5-1 41 41 12 1-2 2 | 2 |

#include <iostream>
#include <vector>
#include <fstream>

#define cin in
#define cout out

using namespace std;

ifstream in("input.txt");
ofstream out("output.txt");

struct toc
{
    int x;
    int y;
};

bool minst(toc a,toc b,toc c)
{
    if (((a.x>=c.x && b.x<=c.x) || (a.x<=c.x && b.x>=c.x)) && ((a.y>=c.y && b.y<=c.y) || (a.y<=c.y && b.y>=c.y)))
        return true;
    return false;
}

int main()
{
    int n;
    int kil=0;
    bool a;
    cin >> n;
    vector <toc> F(n);
    for (int i=0;i<n;i++)
        cin >> F[i].x >> F[i].y;
    for (int i=0;i<n;i++)
    {
        for (int k=i+1;k<n;k++)
        {
            if (!(F[i].x==F[k].x || F[i].y==F[k].y))
            {
            a = true;
            for (int j=0;j<n;j++)
            {
                if (j!=k && j!=i && minst(F[i],F[k],F[j]))
                {
                    a=false;
                    break;
                }
            }
            kil+=a;
            }
        }
    }
    cout << kil;
    return 0;
}