



Использовать только эту сторону листа,
обратная сторона не проверяется!

ШИФР 917

Класс 11 Вариант 8 Дата Олимпиады 17.02.2017

Площадка написания КНЦПЧ

Задача	1	2	3	4	5	6	Σ		Подпись
							Цифрой	Прописью	
Оценка	10	3	9	0	0	1	23	Двадцать три	

10 10 9 0 0 9 38

Двадцать
восемь

Σ 38

Задача №6

```
#include <bits/stdc++.h> // подключение библиотек
using namespace std;

bool isPrime is_prime(int x) { // проверка числа на простоту
    if (x == 2) return true; // 2 - первое простое число
    if (x == 1 || x % 2 == 0) return false; // единица и четные числа не являются простыми
    int i = 3; // будем перебирать все возможные делители числа x
    while (i * i <= x) { // перебираем i в процессе до корня из x
        if (x % i == 0)
            return false;
        i += 2;
    }
    return true; // мы на этапе деления, значит простое
}

int main() {
    int n = 100, sum; bool good = false;
    vector<int> arr(n); int arr[101];
    for (int i = 0; i < n; i++)
        cin >> arr[i];
    cin >> sum;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            for (int k = 0; k < n; k++) {
                for (int l = 0; l < n; l++) {
                    if (i == k || i == j || i == l || j == k || j == l || k == l) // номера попарно различны
                        continue; // на данном этапе равенство
                    if (arr[i] + arr[j] + arr[k] + arr[l] == sum) // если совпадает с sum
                        good = true;
                }
            }
        }
    }
    good = good & is_prime(sum); // для корректности делаем еще проверку
    if (good)
        cout << "Ответ сформирован";
    else
        cout << "Ответ не сформирован";
    cout << "Контрольная сумма " << sum;
    return 0;
}
```

(98.)

Коррентные
не эффециенты по
времени выполнения

Задача 3

При $n = 2$ существует 21 число XYZ: {11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89} ✓

Давиме будем проверять, ~~что~~ перебираем число XYZ так, чтобы любая пара входила в вышеуказанное множество или во множество {02, 03, 05, 07}

// GNU C++ 11

#include <bits/stdc++.h> // подключаем все необходимые библиотеки

using namespace std;

int n, ans = 0; // изначально кол-во подходящих чисел 0

~~vector~~ ~~vector~~

vector<string> good = {"2357", "1373", "39", "17", "137", "39", "17",
"139", "39", "7"};

~~vector~~

// вектор "хороших" пар: для каждой цифры хранить множество
// тех цифр, которые могут находиться после нее

void rec(string s) { // рекурсивный перебор

if (s.size() == n) { // если достигли ~~неизвестной~~ ~~неизвестной~~ длины

ans++;

return;

// обновляем ответ
// и закончим работу рекурсии

int i = s[s.size() - 1] - '0'; // последняя цифра в строке s

for (int j = 0; j < good[i].size(); j++) // перебираем подходящие цифры

rec(s + good[i][j]); // и запускаем рекурсию

}

int main() {

ifstream in("input.txt", "r", stdin); // ~~ввод~~ ~~ввод~~

ofstream out("output.txt", "w", stdout);

cin >> n;

for (int i = 1; i <= 9; i++) // запускаемся от всех цифр ∈ [1; 9],

rec("0" + i);

// т.к. с 0 число начинаться не может

cout << ans; // выводим ответ

return 0;

}

Задача 4

Переберем все возможные варианты K, L, M, N и посчитаем кол-во единиц в результате

K	L	M	N	res	K	L	M	N	res	K	L	M	N	res
0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	1	0	0	1	0	1	1	1	0	0	1	1
0	0	1	0	0	0	1	1	0	1	1	0	1	0	1
0	0	1	1	0	0	1	1	1	1	1	0	1	1	1

Ответ: 9

✓ 108

Задача 5
~~Мы будем пытаться расширить наш набор программных по ширине и высоте~~

Задача 2
Наименьшей рождествой системой счисления является исчерпывающая система счисления. Проверим ее

$$3501_6 = 829_{10}$$

$$3515_6 = 839_{10}$$

$$3541_6 = 853_{10}$$

$$3543_6 = 855_{10}$$

$$3551_6 = 859_{10}$$

$$3555_6 = 863_{10}$$

$$4021_6 = 877_{10}$$

Все значения этого массива - простые числа

Ответ: простое

108.

Задача 4

Полученные данные переводятся в двоичную систему счисления 08.

Задача 5.

#Python 3.3

n = int(input()); maxi = 1 # макс. размер прямоугольника

arr = [0] * n

for i in range(n):

arr[i] = int(a) for a in input().split() # читаем массив

for a in range(n):

for b in range(n):

for c in range(a, n):

for d in range(b, n):

c = 0

for x in range(a, c+1):

for y in range(b, d+1):

if arr[x][y] !=

c += 1

if c == (d - b + 1) * (c - a + 1): # если все клетки закрашены

maxi = max(c, maxi) # учли или не учли ответ

print(maxi)

08.