

ШИФР 8221

Класс 11 Вариант 8 Дата Олимпиады 17.02.17

Площадка написания КНУТУ

| Задача | 1  | 2 | 3 | 4 | 5  | 6 | $\Sigma$ |                  | Подпись                                                                             |
|--------|----|---|---|---|----|---|----------|------------------|-------------------------------------------------------------------------------------|
|        |    |   |   |   |    |   | Цифрой   | Прописью         |                                                                                     |
| Оценка | 10 | 0 | 1 | 0 | 20 | 1 | 32       | Тридцать<br>9 ба |  |

1.  $\Lambda$  - логическое умножение

$V$  - логическое сложение

$$(KVL) \wedge (M \vee N) = 1$$

Это уравнение имеет решение только когда

$$KVL = 1 \quad \text{и} \quad M \vee N = 1$$

$$KVL = 1 \text{ имеет 3 решения}$$

|   |   |
|---|---|
| 1 | 1 |
| 0 | 1 |
| 1 | 0 |

$$M \vee N = 1 \text{ имеет 3 решения}$$

|   |   |
|---|---|
| 1 | 1 |
| 0 | 1 |
| 1 | 0 |

$$3 \cdot 3 = 9$$

Ответ: 9 различных наборов

3. Сначала найдем все простые числа от 10 до 99 и  
X (используя быстрое алгоритмы). Потом посчитаем  
кол-во простых чисел, начинающихся с числа 1 до  
9. Любое число XYZ начинаем с одного из простых  
цифры числа и заканчиваемся. Создаем рекурсивной  
алгоритм.

~~#include <iostream>~~ #include <bits/stdc++.h>

#include <iostream>

using namespace std;

vector<int> a[10];

int ~~ANS~~ = 0;  
int n = 2; int kek = 0;  
int ans(int z) {

~~ANS~~ += a[z].size; if (n == kek) return;

for (int i = 0; i < a[z].size; i++) {

ANS += ans(a[z][i]);

}

int main() {

bool k[100];

for (int i = 2; i < 100; i++) {

if (k[i] == 0) {

~~for (int j = i; j < 100; j++)~~ for (int j = i + 1; j < 100; j++) {

if (j % i == 0)

k[j] = 1;

}

}

}

```
3. for (int i=10; i<20; i++) {
    if (K[i] == 0)
        a[1].push_back(i%10);
}
```

```
for (int i=20; i<30; i++) {
    if (K[i] == 0)
        a[2].push_back(i%10);
}
```

и т.д.

```
for (int i=90; i<100; i++) {
    if (K[i] == 0)
        a[9].push_back(i%10);
}
```

kek = 10;

```
for (int i=0; i<10; i++) for (int i=10; i<99; i++) {
    if (K[i] == 0)
        ans[i%10];
    kek = 10; n = 2;
```

```
cout << ANS << endl;
```

ANS = 0

kek = 10;

n = 2;

```
for (int i=10; i<99; i++) {
    if (K[i] == 0)
        ANS ans[i%10];
    kek = 10; n = 2;
```

3

3.  
cout << "ANS << endl;  
return 0;

}

6.  
include <iostream>  
using namespace std;  
int a[10];  
int n = 0;  
int main()

for (int i = 0; i < 10; i++)  
cin >> a[i];

if (cin >> n)  
for (int i = 0; i < 10; i++)  
for (int j = i + 1; j < 10; j++)

for (int k = j + 1; k < 10; k++)

for (int z = k + 1; z < 10; z++)

if (a[i] + a[j] + a[k] + a[z] == n) {

cout << "Контрольная сумма: " << n << endl;

cout << "Отчет сформирован" << endl;

return 0;

}

cout << "Отчет не сформирован" << endl;

return 0;

}

(Этот алгоритм  
работает за 0.1000)

Это очень быстрый  
алгоритм, очень эффективен  
по времени)

5. В форме угадать пойти и средний  
пути и блок малыши (!!)

```
include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{ int n;
```

```
cin >> n;
```

```
int a[n][n];
```

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

```
for (int j=0; j<n; j++)
```

```
cin >> a[i][j];
```

```
int ans = 0
```

```
int maxx = 0; // если надо пойти блок малыши,
```

м.к. условие некорректно

```
int kol = 0;
```

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

```
for (int j=0; j<n; j++)
```

```
{ if (a[i][j] == 1)
```

```
{ int z = i; int kek = 0;
```

```
int k = j; int lel = 0;
```

```
kol++;
```

while

for

```
while (a[z][j] == 1) while (a[z][k] == 1)
```

```
{ kek++; z++;
```

```
while (a[k][z] == 1) while (a[i][k] == 1)
```

```
{ lel++; k++;
```

```
5. for (int z=i; z<=i+k*k; z++) {
    for (int k=j; k<=j+l*l; k++)
        a[z][k]=0;
}
```

```
ans = k*k*l*l;
if (maxx < k*k*l*l)
    maxx = k*k*l*l;
```

```
}
```

```
}
```

```
{ if (kol==0) kol=1;
  cout << ans / kol << endl; }
```

Если все  
похоже  
среди  
фрагментов

```
(cout << maxx << endl);
```

Если по  
всему  
макс  
популяр

```
return 0;
```

```
}
```

2. Ответ: 0 не является