Contents

1 Section1

1.1 basic

```cpp
// c++ code
#include <bits/stdc++.h>
using namespace std;

int main() {
    // test comment
    cout << "test string\n";
}
```

2 Section2

2.1 thm

- 中文測試
- $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

3 Graph

3.1 C129

```cpp
#include <bits/stdc++.h>

using namespace std;
char oil[100][100] = {0};
int m, n;

void dfs( int i, int j )
{
    oil[i][j] = '*';
    if( oil[i-1][j-1] == '@' )
    {
        if( i-1 >= 0 && j-1 >= 0 )
        {
            oil[i-1][j-1] = '*';
            dfs( i-1, j-1 );
        }
    }
    else if( oil[i-1][j] == '@' )
    {
        if( i-1 >= 0 )
        {
            oil[i-1][j] = '*';
            dfs( i-1, j );
        }
    }
    else if( oil[i-1][j+1] == '@' )
    {
        if( i-1 >= 0 && j+1 <= n )
        {
            oil[i-1][j+1] = '*';
            dfs( i-1, j+1 );
        }
    }
    else if( oil[i][j-1] == '@' )
    {
        if( j-1 >= 0 )
        {
            oil[i][j-1] = '*';
            dfs( i, j-1 );
        }
    }
    else if( oil[i][j+1] == '@' )
    {
        if( j+1 <= n )
        {
            oil[i][j+1] = '*';
            dfs( i, j+1 );
        }
    }
    else if( oil[i+1][j-1] == '@' )
    {
        if( i+1 <= m && j-1 >= 0 )
        {
            oil[i+1][j-1] = '*';
            dfs( i+1, j-1 );
        }
    }
    else if( oil[i+1][j] == '@' )
    {
        if( i+1 <= m )
        {
            oil[i+1][j] = '*';
            dfs( i+1, j );
        }
    }
    else if( oil[i+1][j+1] == '@' )
    {
        if( i+1 <= m && j+1 <= n )
        {
            oil[i+1][j+1] = '*';
            dfs( i+1, j+1 );
        }
    }
}

int main(void)
{
    while( cin >> m >> n )
    {
        int ans = 0;
        if(( m == 0 ) && ( n == 0 ))
        {
            break;
        }
        else
        {
            for( int i = 0 ; i < m ; i++ )
            {
                for(int j = 0 ; j < n ; j++ )
                {
                    cin >> oil[i][j];
                }
            }
        }
        for( int i = 0 ; i < m ; i++ )
        {
            for(int j = 0 ; j < n ; j++ )
            {
                if( oil[i][j] == '@' )
                {
                    dfs( i, j);
                    ans++;
                }
            }
        }
        cout << ans <<endl;
    }
```

```
    return 0;
}
```

3.2 11935

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    int num, flag = 1;
    cin >> num;
    while( num > 0 )
    {
        int n, ans = 0;
        char map[100][100] = {0};
        cin >> n;
        for( int i = 0 ; i < n ; i++ )
        {
            for(int j = 0 ; j < n ; j++ )
            {
                cin >> map[i][j];
            }
        }
        for( int i = 0 ; i < n ; i++ )
        {
            for(int j = 0 ; j < n ; j++ )
            {
                if( map[i][j] == 'x' )
                {
                    ans++;
                }
            }
        }
        cout << "Case " << flag << ": " << ans <<endl;
        num--;
        flag++;
    }
    return 0;
}
```

4 Numbers

4.1 CongruenceEquation

```
#include <bits/stdc++.h>

using namespace std;

long long Mode(long long a, long long n, long long m)
{
    long long sum = 1;
    for( ; n ; n >>= 1 )
    {
        if( n & 1 )
        {
            sum = ( sum * a ) % m;
        }
        a = ( a * a ) % m;
    }
    return sum;
}

int main(void)
{
    int a, b, p, x, ans = 0;
    cin >> a >> b >> p >> x;
    for( int i = 1 ; i < x + 1 ; i++ )
    {
        int n;
        n = i % p;
        n = n * Mode( a, i, p);
        if( n % p == b % p )
        {
            ans++;
        }
    }
    cout << ans <<endl;
    return 0;
}
```