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

using namespace std;


int main()

{
    char binario[100000];
    register int i,j;
    int n,teste,result;
    scanf("%s",binario);
    int tamanho = strlen(binario);
    multiset<int> res;

         scanf("%d",&n);

         for(j=0;j<n;j++)
         {
                  scanf("%d",&teste);
                  result =0;
        for (i = 0; i < tamanho; i++)
            result = (result*2 + (int)binario[i] - '0') %teste;
        if(result==0){
            res.insert(teste);
        }
         }

         int s = res.size();
         if(s==0)
                  printf("Nenhum\n");
         else {
                  i=0;
                  for (auto k : res){
                           printf("%d%c", k, i == s-1 ? '\n' : ' ');
                           i++;
                  }
         }
         return 0;
}
```

```cpp
#include <iostream>

using namespace std;

struct v{
    int bff[1001];
    int qtdBff=0;
};

bool visitas[1001];

v alunos[1001];
int qtdALunos;
int num;
int grupos=0;
bool alone=false;

void rec(int now)
{
    visitas[now]=true;
    for(int i=0;i<alunos[now].qtdBff;i++)
    {
        if(!visitas[alunos[now].bff[i]])
            rec(alunos[now].bff[i]);
    }
}

int main()
{
    int I,J;
    cin >> qtdALunos;
    cin >> num;

    for(int i=1;i<=num;i++)
    {
        cin >> I;
        cin >> J;
        alunos[I].bff[alunos[I].qtdBff++] = J;
        alunos[J].bff[alunos[J].qtdBff++] = I;
    }
    for(int i=1;i<=qtdALunos;i++)
    {
        if(alunos[i].qtdBff==0){
            grupos++;
        }
        else if(!visitas[i])
        {
            grupos++;
            rec(i);
        }
    }
    cout << grupos<<endl;

    return 0;
}
```

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

using namespace std;

int amod(string num, int a)
{
    int res = 0;

    for (int i = 0; i < num.length(); i++)
         res = (res*2 + (int)num[i] - '0') %a;

    return res;
}

int main()

{
    string binario;
    int n,i,j,teste,result;
    cin >> binario;

    multiset<int> res;

         scanf("%d",&n);

         for(j=0;j<n;j++)
         {
                  scanf("%d",&teste);
        result = amod(binario,teste);
        if(result==0){
            res.insert(teste);
        }
         }

         int s = res.size();
         if(s==0)
                  printf("Nenhum\n");
         else {
                  i=0;
                  for (auto j : res){
                           printf("%d%c", j, i == s-1 ? '\n' : ' ');
                           i++;
                  }
         }
         return 0;
}
```

```cpp
#include <iostream>

using namespace std;

struct v
{
    int bff[1001];
    int qtdBff=0;
};

bool visitas[1001];

v alunos[1001];
int qtdALunos;
int num;
int total;
int me;

bool alone=false;

void rec(int now)
{
    visitas[now]=true;
    total++;
    for(int i=0; i<alunos[now].qtdBff; i++)
    {
        if(!visitas[alunos[now].bff[i]])
            rec(alunos[now].bff[i]);
    }
}

int main()
{
    int I,J;

    while(scanf("%d",&qtdALunos)==1)
    {

        cin >> num;
        total=0;

        for (int i=1;i<=qtdALunos ;i++ )
        {
            alunos[i].qtdBff=0;
            visitas[i]=false;
        }


        for(int i=1; i<=num; i++)
        {
            cin >> I;
            cin >> J;
            alunos[I].bff[alunos[I].qtdBff++] = J;
            alunos[J].bff[alunos[J].qtdBff++] = I;
        }
        cin >> me;
        rec(me);

        cout << total<<endl;
    }
    return 0;
}
```

```
#include <iostream>
#include <cstdlib>
#include <cstdio>
using namespace std;

int main()
{
    int qtdS,qtdK;
    bool flag=false;
    char str[100001];
    char strK[1001];
    scanf("%d",&qtdS);
    getchar();

    for (int i=0;i<qtdS ;i++ )
    {
        gets(str);
        scanf("%d",&qtdK);
        getchar();


        for (int j=0;j<qtdK ;j++ )
        {
            gets(strK);
            for (int k=0;k<100001 ;k++ )
            {
                if(str[k]=='\0')
                {
                    cout << "n" << endl;
                    break;
                }
                if(str[k]==strK[0])
                {
                    flag=true;
                    for (int l=1;l<1000 ;l++ )
                    {
                        if(strK[l]=='\0')
                            break;
                        if(str[k+l]!=strK[l])
                        {
                            k=k+l-1;
                            flag=false;
                            break;
                        }

                    }
                    if(flag)
                    {
                        cout << "y" << endl;
                        break;
                    }
                }
            }

        }
    return 0;
}
```

```
 1entrada = str(input())
 2num = int(str(entrada), 2)
 3qtd = int(input())
 4lista = []
 5for i in range(qtd):
 6    divisor = int(input())
 7    if(num%divisor==0):
 8        lista.append(divisor)
 9
10if(len(lista)==0):
11    print("Nenhum")
12else:
13    lista.sort()
14    print (' '.join(map(str, lista)))
```

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

using namespace std;

long long int getSum(int BITree[], int index)
{
    long long int sum = 0; // Initialize result

    // Traverse ancestors of BITree[index]
    while (index > 0)
    {
        // Add current element of BITree to sum
        sum += BITree[index];

        // Move index to parent node in getSum View
        index -= index & (-index);
    }
    return sum;
}

// Updates a node in Binary Index Tree (BITree) at given index
// in BITree.  The given value 'val' is added to BITree[i] and
// all of its ancestors in tree.
void updateBIT(int BITree[], int n, int index, int val)
{
    // Traverse all ancestors and add 'val'
    while (index <= n)
    {
        // Add 'val' to current node of BI Tree
        BITree[index] += val;

        // Update index to that of parent in update View
        index += index & (-index);
    }
}

// Returns inversion count arr[0..n-1]
long long int getCount(int arr[], int n)
{
    long long int invcount = 0; // Initialize result

    // Find maximum element in arr[]
    int maxElement = 0;
    for (int i=0; i<n; i++)
        if (maxElement < arr[i])
            maxElement = arr[i];

    // Create a BIT with size equal to maxElement+1 (Extra
    // one is used so that elements can be directly be
    // used as index)
    int BIT[maxElement+1];
    for (int i=1; i<=maxElement; i++)
        BIT[i] = 0;

    // Traverse all elements from right.
    for (int i=0; i<=n-1; i++)
    {
        // Get count of elements smaller than arr[i]
        invcount += getSum(BIT, arr[i]-1);

        // Add current element to BIT
        updateBIT(BIT, maxElement, arr[i], 1);
    }

    return invcount;
}

int main()
{

    int n;
    int arr[100000];
    while(scanf("%d",&n)==1)
    {
        for (int i=0; i<n ; i++ )
        {
            scanf("%d",&arr[i]);
        }
        printf("%lld\n",getCount(arr,n));
    }
```

```
82    return 0;
83}
```

```
 1#include <iostream>
 2#include <algorithm>
 3#include <math.h>
 4#include <stack>
 5#include <vector>
 6#define PI 3.141592653
 7
 8using namespace std;
 9
10typedef struct ponto
11{
12    double x,y;
13    float angulo;
14} ponto;
15
16   vector<ponto> p;
17    vector<ponto> hull;
18
19ponto nP(int x, int y) {
20    ponto p;
21    p.x = x;
22    p.y = y;
23    return p;
24}
25
26double dist(ponto A,ponto B)
27{
28    return sqrt(pow(A.x-B.x,2)+pow(A.y-B.y,2));
29}
30
31bool operator<(const ponto& a, const ponto& b)
32{
33    return a.angulo < b.angulo;
34}
35
36float ccw(ponto p1,ponto p2,ponto p3)
37{
38    return (p2.x - p1.x)*(p3.y - p1.y) - (p2.y - p1.y)*(p3.x - p1.x);
39}
40
41float anglePa(int x1,int y1,int x2,int y2)
42{
43    return abs((atan2( y2-y1,x2-x1 ))* 180 /PI);
44}
45
46float angulo(ponto A,ponto B)
47{
48    return anglePa(A.x,A.y,B.x,B.y);
49}
50
51ponto next_to_top(stack<ponto> pontos)
52{
53    ponto aux = pontos.top();
54    pontos.pop();
55    ponto next = pontos.top();
56    pontos.push(aux);
57    return next;
58}
59
60int main()
61{
62    int qtd,tY,tX,iM;
63    int menorY;
64    double mais=0.0,menos=0.0;
65    double area;
66    stack<ponto> pontos;
67
68    //ponto p[100001];
69
70
71    scanf("%d",&qtd);
72    p.clear();hull.clear();
73    hull.reserve(qtd);
74    p.reserve(qtd);
75    pontos=stack<ponto>();
76    mais=0.0;menos=0.0;
77    menorY=10001;
78    iM=0;
79    for (int i=0; i<qtd ; i++)
80    {
81        cin >> tX >> tY;
```

```
 82
 83        if(tY<menorY)
 84        {
 85            menorY=tY;
 86            iM=i;
 87        }
 88        p.push_back(nP(tX,tY));
 89    }
 90    swap(p[iM],p[0]);
 91    p[0].angulo=0;
 92    for (int i=1; i<qtd ; i++ )
 93    {
 94        p[i].angulo = angulo(p[0],p[i]);
 95    }
 96
 97    //sort(p+1,p+qtd,[](ponto pA,ponto pB){return pA.angulo > pB.angulo;});
 98    sort(p.begin()+1,p.end());
 99
100
101    for (int i=1; i<p.size()-1 ; i++ )
102    {
103        if(p[i].angulo==p[i+1].angulo){
104            if(dist(p[i],p[0])>dist(p[i+1],p[0]))
105            {
106                p.erase(p.begin()+i+1);
107            }
108            else
109            {
110                p.erase(p.begin()+i);
111            }
112            if(p[i].angulo==p[i+1].angulo)
113                i--;
114        }
115
116    }
117
118  /*  for (int i=0; i<p.size() ; i++ )
119    {
120        cout << p[i].x<<", "<<p[i].y <<endl;
121    }*/
122
123
124
125    pontos = stack<ponto>();
126
127    pontos.push(p[0]);
128    pontos.push(p[1]);
129    pontos.push(p[2]);
130    for (int i=3; i<p.size() ; i++)
131    {
132        while(ccw(next_to_top(pontos),pontos.top(),p[i])<=0)
133            pontos.pop();
134        pontos.push(p[i]);
135    }
136
137
138
139    while(pontos.size())
140    {
141        //cout << pontos.top().x<<", "<<pontos.top().y<<endl;
142        hull.push_back(pontos.top());
143        pontos.pop();
144    }
145
146    mais += hull[hull.size()-1].x*hull[0].y;
147    menos += hull[0].x*hull[hull.size()-1].y;
148
149    for (int i=0;i<hull.size()-1 ;i++ )
150    {
151        mais += hull[i].x*hull[i+1].y;
152        menos += hull[i+1].x*hull[i].y;
153    }
154
155    area = abs(mais-menos)/2.0;
156    printf("%.2f\n",area);
157
158    return 0;
159}
```

```cpp
#include <iostream>
#include <math.h>
using namespace std;

int main()
{
    int n;
    int v;
    while(true)
    {
        cin >> n;
        if(n==0)
            break;
        else
        {
            cout << 1 ;
            for(int i=2; ; i++)
            {
                v = i*i;
                if(v<=n)
                    cout << " "<<v;
                else
                    break;
            }

            cout << endl;
        }


    }

    return 0;
}
```

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
	long long int N, B;

	scanf("%lld %lld", &N, &B);

	long long int ini = 0, fim = N / 2, meio, X = 0;

	while (ini <= fim)
	{
		meio = (ini + fim) / 2;
		if (4 * meio * (N - meio) < B)
		{
			X = meio;
			ini = meio + 1;
		}
		else
			fim = meio - 1;
	}

	long long int resto_feijoes = B - 4 * X * (N - X);
	long long int casca = X + 1;
	long long int lim = N + (casca - 1) * (-2);
	long long int coordX = casca, coordY = casca;
	long long int resto;


	if(resto_feijoes >= lim)
	{
		resto_feijoes -= lim;
		coordY += (lim - 1);
	}
	else
	{
		resto = resto_feijoes % lim;
		coordY += (resto - 1);
		resto_feijoes -= resto;
	}

	if(resto_feijoes > 0)
	{
		if(resto_feijoes >= lim - 1)
		{
			resto_feijoes -= (lim - 1);
			coordX += (lim - 1);
		}
		else
		{
			resto = resto_feijoes % (lim - 1);
			coordX += resto;
			resto_feijoes -= resto;
		}

		if(resto_feijoes > 0)
		{
			if(resto_feijoes >= lim - 1)
			{
				resto_feijoes -= (lim - 1);
				coordY -= (lim - 1);
			}
			else
			{
				resto = resto_feijoes % (lim - 1);
				coordY -= resto;
				resto_feijoes -= resto;
			}

			if(resto_feijoes > 0)
				coordX -= resto_feijoes;
		}
	}

	printf("%lld %lld\n", coordX, coordY);

	return 0;
}
```

```
 1#include <iostream>
 2#include <vector>
 3using namespace std;
 4int qtdF;
 5int maxC;
 6
 7struct caso
 8{
 9    int nC,nD;
10};
11
12int m[50][50];
13
14vector<caso> frase;
15
16
17int rec(int f, int sD,int sC)
18{
19    int temp=0;
20    int record=0;
21    if(frase[f].nC+sC<=maxC)
22    {
23        sD+=frase[f].nD;
24        sC+=frase[f].nC;
25        if(f==qtdF-1)
26            return sD;
27        for(int i=f+1; i<qtdF; i++)
28        {
29            temp = rec(i,sD,sC);
30            if(temp>record)
31                record = temp;
32        }
33        return record;
34
35    }
36    else
37    {
38        return sD;
39    }
40
41}
42
43int main()
44{
45    frase.reserve(50);
46    int record=0,temp,cont=1,tC,tD;
47    while(true)
48    {
49        record =0;
50        cin >> maxC;
51        cin >> qtdF;
52
53        if(maxC==0)
54            break;
55
56        for(int i=0; i<qtdF; i++)
57        {
58            cin >> tC;
59            cin >> tD;
60            if(tC<=maxC)
61            {
62                frase[i].nC = tC;
63                frase[i].nD = tD;
64            }
65            else
66            {
67                qtdF--;
68            }
69        }
70        for(int i=0; i<qtdF; i++)
71        {
72            temp = rec(i,0,0);
73            if(temp>record)
74                record = temp;
75        }
76        cout << "Teste " << cont++<<endl;
77        cout << record << endl<<endl;
78
79    }
80
81
```

```
82    return 0;
83}
```