1.Number of steps

```
#include<stdio.h>
int main()
{
    int n,k,steps=0;
    scanf("%d",&n);
    int i,a[n],b[n];
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n;i++)
        scanf("%d",&b[i]);
    for(i=0;i<n-1;i++)
    {
        if(a[i]<a[i+1])
        {
            k=a[i];
            a[i]=a[i+1];
            a[i+1]=k;
            k=b[i];
            b[i]=b[i+1];
            b[i+1]=k;
        }
    }
    for(i=0;i<n-1;i++)
    {
        while(a[n-1]!=a[i])
        {
            if(a[i]<=0)
            {
                printf("-1");
                exit(0);
            }
            if(a[n-1]<a[i])
            {
                a[i]=a[i]-b[i];
                steps++;
            }
            if(a[n-1]>a[i])
            {
                a[n-1]=a[n-1]-b[n-1];
                steps++;
            }
        }

    }
    printf("%d",steps);
    return 0;
}
```

2. Zoos

```
#include<string.h>
#include<stdio.h>
int main()
```

```
{
 char s[20];
 int n=0,t=0,i;
 gets(s);
 for(i=0;i<strlen(s);i++)
  {
if(s[i]=='z')
t++;
else if(s[i]=='o')
n++;
else
continue;
  }
 if((t*2)==n)
 printf("Yes");
 else
 printf("No");
 return 0;
}
```

3. Cost of balloons

```
#include<stdio.h>
 int main()
  {
    int t,n,g,p,a[10][2],sum1,sum2,ans;
    scanf("%d",&t);
    for(int i=0;i<t;i++)
    {
        sum1=sum2=0;
        scanf("%d%d",&g,&p);
        scanf("%d",&n);
        for(int j=0;j<n;j++)
        {
            for(int k=0;k<2;k++)
              scanf("%d",&a[j][k]);
        }
        for(int j=0;j<n;j++)
        {
                if(a[j][0]==1)
                 sum1=sum1+g;
                if(a[j][1]==1)
                 sum1=sum1+p;
        }
        for(int j=0;j<n;j++)
        {
                if(a[j][0]==1)
                 sum2=sum2+p;
                if(a[j][1]==1)
                 sum2=sum2+g;
        }
        ans=sum1>sum2?sum2:sum1;
        printf("%d\n",ans);
    }
 }
```

4. Seven-Segment Display

```
#include <stdio.h>
#include <string.h>

int main()
{
 char N[100+1];
 int T, N_len, i, j, num_sticks;
 int digit_sticks[] = {
  6, 2, 5, 5, 4, 5, 6, 3, 7, 6
 };

 scanf("%d", &T);

 for (i = 0; i < T; i++)
 {
  scanf("%100s", N);
  N_len = (int) strlen(N);
  for (j = 0, num_sticks = 0; j < N_len; j++)
  {
   num_sticks += digit_sticks[N[j] - '0'];
  }
  if ((num_sticks % 2) != 0)
  {
   printf("7");
   num_sticks -= 3;
  }
  for(j = 0; j < (num_sticks/2); j++)
  {
   printf("1");
  }
  printf("\n");
 }

 return 0;
}
```

5. Ali and Helping innocent people

```
#include<stdio.h>

int main()

{

    char a[9];

    scanf("%s",&a);

    if(a[2]!='A'&& a[2]!='E'&& a[2]!='I'&& a[2]!='O'&& a[2]!='U'&& a[2]!='Y'&& (a[0]+a[1])%2==0 && (a[3]+a[
4])%2==0 && (a[4]+a[5])%2==0 && (a[7]+a[8])%2==0){

    printf("valid");}
    else{

    printf("invalid");}




    return 0;

}
```

6. Best Index

```
#include<stdio.h>
#include<math.h>
main()
{
long n,i,k,j,left,p=0,max=-10000000;
scanf("%ld",&n);
long a[n],sum;
for(i=0;i<n;i++)
{
scanf("%ld",&a[i]);
if(i>0)
a[i]+=a[i-1];
}
for(i=0;i<n;i++)
{
left=n-i;
sum=0;
k=(-1+(int)sqrt((double)(8*left+1)))/2;
sum=a[(k*(k+1))/2+i-1];
if(i!=0)
sum-=a[i-1];
if(max<sum)
max=sum;
}
printf("%ld",max);
}
```

7. Toggle String

```
#include<string.h>
void main()
{
 char s[100],c;
 int i,x,l;
 scanf("%s",s);
 l=strlen(s);
 for(i=0;i<=l-1;i++)
 {
  x=s[i];
  if(x<=90)
  x=x+32;
  else
  x=x-32;
  c=x;
  s[i]=c;
 }
 printf("%s",s);
}
```

8. Find Product

```
#include <stdio.h>
int  main(){
 int num;
 int arr[1000];
 scanf("%d",&num);
 long int answer=1;
 for(int i=0;i<num;i++){
  scanf("%d",&arr[i]);
  answer=answer*arr[i]%(1000000007);

 }

 printf("%ld",answer);
}
```

9. Palindromic String

```
#include<stdio.h>
#include<string.h>
void main(){
 char str[1000];
 int i,j,k,count=0;
 scanf("%s",str);
 k=strlen(str);
 for(i=0;i<k;i++)
 {
  if(str[i]==str[k-(i+1)])
  count++;

 }
 if(count==k){
  printf("YES");
 }
 else{
  printf("NO");
 }
}
```

10. Factorial!

```
#include <stdio.h>

int main(){
 int num,i,res=1;
 scanf("%d", &num);
```

```
    for(i=1;i<=num;i++)
    res=res*i;
 printf("%d",res);
 return 0;
}
```

11. Life, the Universe, and Everything

```
#include <stdio.h>

int main()
{
    int n;
    while(1)
    {
     scanf("%d",&n);
     if(n==42)
     break;
     printf("%d\n",n);
    }
    return 0;
}
```

IMPLEMENTATIONS :

1. A special number

```
#include<stdio.h>
int main()
{
 int T,i,j,count=0,x,sum=0;
 scanf("%d",&T);
 int a[T];
 for(i=0;i<T;i++)
 {
  scanf("%d",&a[i]);
 }
 for(i=0;i<T;i++)
 {
  count=0;
  for(j=a[i];count==0;j++)
  {
   int c=j;
   sum=0;
   while(c>0)
   {
    x=c%10;
    sum+=x;
    c=c/10;
   }
   if(sum%4==0)
   {
    count++;
    printf("%d\n",j);
   }

  }
 }
 return 0;
}
```

2. Excursion

```
#include <stdio.h>
#include <math.h>

int main()
{
 int t, n, m, k;
 scanf(" %d", &t);

 for(int i = 0; i < t; i++)
 {
  scanf(" %d %d %d", &n, &m, &k);
  int res = (int)ceil((double)n / k) + (int)ceil((double)m / k);
  printf("%d\n", res);
 }

 return 0;
}
```

3. Special matrix

```
#include <stdio.h>
#include <math.h>

#define MAX 1000001
#define MAX_DIVISORS 2000002
#define PRIME_MARKED(A) (prime_marked[A]==0? 1: prime_marked[A])
int prime_marked[MAX_DIVISORS];
int sum_no_prime_divisors[MAX_DIVISORS];

int number_of_prime_divisor(int num) {
 if (num == 0) {
  return 0;
 }

 if (num == 1) {
  return 0;
 }

 if (num == 2) {
  return 1;
 }

 int divisors = 0;
 if (num%2 == 0) {
  divisors++;
 }
 while (num%2 == 0) {
  num /= 2;
 }

 if (num == 1) {
  return divisors;
 }

 for (int i=3; i<sqrt(num); i += 2) {
  if (num%i == 0) {
   divisors++;
  }
  while (num%i == 0) {
   num /= i;
  }
 }

 if (num > 2) {
  divisors++;
 }
 return divisors;
}

void mark_prime_numbers(int max_divs) {
 for (int i=2; i<max_divs/2; i++) {
  for (int j=2; j*i<max_divs; j++) {
   if (prime_marked[i] == 0) {
    prime_marked[i*j]++;
   }
  }
 }
}


void prepare() {
 mark_prime_numbers(MAX_DIVISORS);
}

int main() {
 prepare();
 int tst;
 scanf("%d", &tst);
 while (tst--) {
  int n, m;
  unsigned long long sum = 0;
  scanf("%d %d", &n, &m);
  if (n==m) {
   int k = 2;
   for (int i=1; i<=n; i++) {
    sum += PRIME_MARKED(k)*i;
    k++;
   }
   for (int i=n-1; i>=1; i--) {
    sum += PRIME_MARKED(k)*i;
    k++;
   }
  } else {
   int k = 2, min, max, i;
   min = (n<m? n: m);
   max = (n>m? n: m);

   for (i=1; i<=min; i++) {
    sum += PRIME_MARKED(k)*i;
    k++;
   }

   while (i<=max) {
    sum += PRIME_MARKED(k)*min;
    k++;
    i++;
   }

   for (i=min-1; i>=1; i--) {
    sum += PRIME_MARKED(k)*i;
    k++;
   }
  }
  printf("%llu\n", sum);
 }
 return 0;
}
```

4. Lunch boxes

```
#include <stdio.h>
int partition(int arr[], int left, int right)

{

    int i = left, j = right;

    int tmp;

    int pivot = arr[(left + right) / 2];



    while (i <= j) {

        while (arr[i] < pivot)

            i++;

        while (arr[j] > pivot)

            j--;

        if (i <= j) {
```

```
                  tmp = arr[i];

                  arr[i] = arr[j];

                  arr[j] = tmp;

                  i++;

                  j--;

            }

      };



      return i;

}



void quickSort(int arr[], int left, int right) {

      int index = partition(arr, left, right);

      if (left < index - 1)

            quickSort(arr, left, index - 1);

      if (index < right)

            quickSort(arr, index, right);

}

int main(){
 unsigned int num;
 scanf("%u", &num);
   for(int i=0;i<num;i++){
        unsigned int ms,x;
   unsigned int nb;
   scanf("%u %u", &nb,&ms);
          unsigned int so[ms];
   for(int j =0;j<ms;j++){
    scanf("%u", &so[j]);
   }
   quickSort(so,0,ms-1);
    int temc=0,c=0;
   for(int k=0;k<ms;k++){
    if(temc < nb){
     temc=temc+so[k];
     if(temc <= nb)
     c++;
    }else{
     break;
    }
   }

   printf("%d\n",c);


   }
 }
```

5. Erasing an array

```
#include<stdio.h>
int main()
{
int t;
scanf("%d",&t);
while(t--){
int n;
int count=1;
scanf("%d",&n);
int arr[n];
for(int i=0;i<n;i++){
scanf("%d",&arr[i]);
}
for(int i=0;i<n-1;i++){
if(arr[i]==1 && arr[i+1]==0)
count++;
}
    printf("%d\n",count);
}
}
```

6. Path queries

```
#include <stdio.h>
```

```
int main()
{
    long long int t;
    scanf("%lld",&t);

    while (t--){
        long long int n, q;
        scanf("%lld" "%lld", &n, &q);

        long long int a[n];
        long long int odd=0;
        long long int even=0;

        for (int i=1; i<=n; i++){
            scanf("%lld", &a[i]);
            if (a[i]%2){
                odd++;
            } else {
                even++;
            }
        }

        for (long long int i=1; i<n; i++){
            long long int a, b;
            scanf("%lld" "%lld", &a, &b);
        }

        while (q--){
            long long int i, val,sum;
            scanf("%lld" "%lld", &i, &val);

            if (a[i]%2){
                odd--;
            } else {
                even--;
            }

            a[i] = val;
            if (a[i]%2){
                odd++;
            } else {
                even++;
            }

            sum = odd*(odd+1)/2 + even*(even+1)/2;

            printf("%lld ", sum);
        }

        printf("\n");
    }

    return 0;
}
```

7. Simon cannot sleep

```
#include <stdio.h>

int main(){
 char time[5];
 int h,m;
 int count;
 scanf("%s",time);

 // for hour
 switch(time[0])
 {
  case '0' : h=0;
           break;
  case '1' : h=1;
           break;
  case '2' : h=2;
           break;
 }
 switch(time[1])
 {
  case '0' : h = h*10 + 0;
           break;
  case '1' : h = h*10 + 1;
           break;
  case '2' : h = h*10 + 2;
           break;
  case '3' : h = h*10 + 3;
           break;
  case '4' : h = h*10 + 4;
           break;
  case '5' : h = h*10 + 5;
           break;
  case '6' : h = h*10 + 6;
           break;
  case '7' : h = h*10 + 7;
           break;
  case '8' : h = h*10 + 8;
           break;
  case '9' : h = h*10 + 9;
           break;
 }

 // for minutes
 switch(time[3])
 {
  case '0' : m=0;
           break;
  case '1' : m=1;
           break;
  case '2' : m=2;
           break;
  case '3' : m=3;
           break;
  case '4' : m=4;
           break;
  case '5' : m=5;
           break;
 }
 switch(time[4])
 {
  case '0' : m = m*10 + 0;
           break;
  case '1' : m = m*10 + 1;
           break;
  case '2' : m = m*10 + 2;
           break;
  case '3' : m = m*10 + 3;
           break;
  case '4' : m = m*10 + 4;
           break;
  case '5' : m = m*10 + 5;
           break;
  case '6' : m = m*10 + 6;
           break;
  case '7' : m = m*10 + 7;
           break;
  case '8' : m = m*10 + 8;
           break;
  case '9' : m = m*10 + 9;
           break;
 }

 // for count
 switch(h)
 {
  case 0 : count = 1;
           break;
  case 1 : count = h;
         if(m >= h*5 + 1)
           count++;
         break;
  case 2 : count = h;
         if(m >= h*5 + 1)
           count++;
           break;
  case 3 : count = h;
         if(m >= h*5 + 2)
           count++;
           break;
  case 4 : count = h;
         if(m >= h*5 + 2)
           count++;
           break;
  case 5 : count = h;
         if(m >= h*5 + 3)
           count++;
           break;
  case 6 : count = h;
```

```
        if(m >= h*5 + 3)
          count++;
          break;
case 7 : count = h;
        if(m >= h*5 + 4)
          count++;
          break;
case 8 : count = h;
        if(m >= h*5 + 4)
          count++;
          break;
case 9 : count = h;
        if(m >= h*5 + 5)
          count++;
          break;
case 10 : count = h;
        if(m >= h*5 + 5)
          count++;
        break;
case 11 : count = h;
        break;
case 12 : count = h;
          break;
case 13 : count = h-1;
        if(m >= 1*5 + 1)
          count++;
          break;
case 14 : count = h-1;
        if(m >= 2*5 + 1)
          count++;
          break;
case 15 : count = h-1;
        if(m >= 3*5 + 2)
          count++;
          break;
case 16 : count = h-1;
        if(m >= 4*5 + 2)
          count++;
          break;
case 17 : count = h-1;
        if(m >= 5*5 + 3)
          count++;
          break;
case 18 : count = h-1;
        if(m >= 6*5 + 3)
          count++;
          break;
case 19 : count = h-1;
        if(m >= 7*5 + 4)
          count++;
          break;
case 20 : count = h-1;
        if(m >= 8*5 + 4)
          count++;
        break;
```

```
  case 21 : count = h-1;
          if(m >= 9*5 + 5)
            count++;
          break;
  case 22 : count = h-1;
          if(m >= 10*5 + 5)
            count++;
            break;
  case 23 : count = h-1;
            break;
 }


 printf("%d",count);
}
```

8. Digit cube

```
#include <stdio.h>
unsigned long long int ndigit(unsigned long long int d){
   unsigned long long int sum=0,p;
 while(d!=0){
     sum=sum+(d%10);
  d=d/10;
 }
 p=sum * sum * sum;
   return p;
}
int main()
{
    unsigned long long int n,t,t1,st[4];
    int num;
 scanf("%d",&num);
 for(int j=0;j<num;j++){
        st[0]=0;
        st[1]=0;
    scanf("%llu %llu",&n,&t1);
    for(int i=0;i<t1;i++){
        t=ndigit(n);
        n=t;
        st[i%4]=t;
        if(st[0]==st[2] && st[1]==st[3]){
            printf("%llu\n",st[3-(t1%4)]);
            goto haha;
        }
    }
    printf("%llu\n",t);
    haha :
    continue;
 }
    return 0;
}
```

10. Robotic moves

```
#include<stdio.h>
main()
{
 int T;
 long long int N;
 scanf("%lld",&T);
 int i;
 for(i=0;i<T;i++)
 {
  scanf("%lld",&N);
  printf("%lld\n",N*(N+1));
 }
}
```

11. Mathematically beautiful numbers

```
#include <stdio.h>

int main()
{
    int T, k, flag, i, rem;
    long long int x;
    scanf("%d", &T);

    for(i=0; i<T; i++)
    {
        scanf("%lld %d",&x , &k );
        flag = 0;

        while(x)
        {
            rem = x % k ;
            if(rem != 0 && rem != 1)
            {
                flag = 1;
                break;
            }
            x = x/k ;
        }

        if(flag == 1)
            printf("NO\n");
        else
            printf("YES\n");
    }
}
```

12. Multiple occurrences

```
#include<stdio.h>
#define MAX 200000
void quickSort(int r[MAX],int z[MAX],int i,int j){
    int smaller=i+1,larger=j,swap,pivot,k;
    if(i>=j) return;
    while(smaller<larger){
        if(r[smaller]<=r[i]) smaller++;
        else{
            if(r[larger]>r[i]) larger--;
            else {
                swap=r[smaller];
                r[smaller]=r[larger];
                r[larger]=swap;
                swap=z[smaller];
                z[smaller]=z[larger];
                z[larger]=swap;
            }
        }
    }
    if(r[smaller]<=r[i]) pivot=smaller;
    else pivot=smaller-1;
    swap=r[i];
    r[i]=r[pivot];
    r[pivot]=swap;
    swap=z[i];
    z[i]=z[pivot];
    z[pivot]=swap;
    quickSort(r,z,i,pivot-1);
    quickSort(r,z,pivot+1,j);
}
void main(){
    int a[MAX],b[MAX],t,n,i,j,sum,pivot,max,min;
    scanf("%d",&t);
    while(t--){
        scanf("%d",&n);
        for(i=0;i<n;i++){
            scanf("%d",&a[i]);
            b[i]=i;
        }
        quickSort(a,b,0,n-1);
        pivot=0;
        sum=0;
        min=-1;
        max=-1;
        for(i=0;i<n;i++){
            if(a[i]==a[pivot]){
                if(min==-1&&max==-1){
                    min=b[i];
                    max=b[i];
                }else if(b[i]<min) min=b[i];
                else if(b[i]>max) max=b[i];
            }
            if(a[i]!=a[pivot]){
                sum+=(max-min);
                pivot=i;
                min=b[pivot];
                max=b[pivot];
            }
        }
        sum+=(max-min);
        printf("\n%d",sum);
    }
}
```

13. Anti-palindrome strings

```
#include<stdio.h>
#include<string.h>
int main()
{
 int T=0,i=0,f=0;
 scanf ("%d",&T);
 while(T>0)
 {
  char s[200000];
  int a[26]={};
  scanf ("%s",s);
  int l=strlen(s);
  for(i=0;i<l;++i)
  {
   a[s[i]-97]+=1;

  }
  i=0;
  for(int j =0;j<26;++j)
  {
   while(a[j]>0)
   {
    s[i++]=j+97;
    --a[j];
   }
  }
  for(int i=0;i<l/2;++i)
  {
   if(s[i]!=s[l-1-i])
   {
    printf ("%s\n",s);
    f=1;
    break;
   }
   else
   f=0;
  }
  if(f==0)
  printf ("-1\n");
  --T;
 }
}
```

14. Summation program

```
int main()
{
 int u;
 scanf("%d",&u);
 for(int g=u;g>0;g--)
 {
  long long int j,o,as=0,q;
  scanf("%lld",&o);
  for(int z;z<1000;z++);
  q=sqrt(o);
  for(j=1;j<=q;j++)
     as+=2*(o/j);
  printf("%lld\n",as-q*q);
 }
 return 0;
}
```

15. Special numbers

```
int n, k = 0;
int m[1000000];
int gcd(int fD, int sD) {
  if (fD == sD)
  return fD;
  if (fD == 0)
  return sD;
  return gcd(sD % fD, fD);
}
void f(long long int v) {
  if (v > n) return;
  if (v > 0) m[k++] = v;
  f(10 * v + 4);
  f(10 * v + 7);
}
int main() {
  scanf("%d", &n);
  f(0);
  int result = 0;
  for (int i = 0; i < k; i++)
    for (int j = i + 1; j < k; j++)
      if (gcd(m[i], m[j]) == 1) result++;
  printf("%d\n", result);
  return 0;
}
```

16. Finding vaccines

```
#include<stdio.h>
int main()
{
    int n, n1, c=0, g=0,lop=0,in=0,pos=0;
    scanf("%d",&n);
    scanf("%d",&n1);
    char a[1000];
    scanf("%s", a);
    for(int i =0; i < n1; i++)
    {
        if(a[i]=='G')
        {
            g++;
        }
        if(a[i]=='C')
        {
            c++;
        }
    }
    while(n--)
    {
        int l, C=0, G=0,count=0;
        char v[1000];
        scanf("%d",&l);
        scanf("%s", v);
        for(int i =0; i < l; i++)
        {
            if(v[i]=='G')
            {
                G++;
            }
            if(v[i]=='C')
            {
                C++;
            }
        }
        count=G*c+C*g;
        pos++;
        if(count>lop)
        {
            lop=count;
            in=pos;
        }
    }
    printf("%d",in);
    return 0;
}
```

17. The largest subnumber

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#include<limits.h>
#define max(a,b) (a>b?a:b)
#define min(a,b) (a<b?a:b)
#define endl printf("\n")
#define input(x) scanf("%d",&x)
#define inputll(x) scanf("%lld",&x)
#define inputc(x) scanf("%c",&x);
#define inputstr(x) scanf("%s",&x);
#define and &&
#define or ||
#define ll long long
ll arr[100005];

int main(){
    int t;input(t);
    while(t--){
        ll n,k;
        inputll(n);inputll(k);
        arr[0]=1;
        for(int i=1;i<=100000;i++)arr[i] =(arr[i-1]*10)%k;
        char s[n];inputstr(s);
        ll prefix[n];prefix[0] = s[0]-'0';
        for(int i=1;i<n;i++)prefix[i] = prefix[i-1]^(s[i]-'0');
        int idx =-1;ll rem =0,maxm=0;
        for(int i=n-1;i>0;i--){
            rem=((s[i]-'0')*arr[n-i-1] + rem)%k;
            if(!rem and s[i]!='0' and prefix[i-1]>=maxm){
                idx=i;
                maxm = prefix[i-1];
            }
        }
        if(idx == -1)printf("-1");
        else for(int i=idx;i<n;i++)printf("%c",s[i]);
        endl;
    }
    return 0;
}
```

18. Supernatural

```
#include<stdio.h>

int main()

{

  int n ,sum =1,c=0;




  scanf("%d",&n);



for(int i = 1 ; i<=322222; i++)

{

   if(i==100)

     i = i + (10*2);

else if(i==1000)

     i = i + (2*10*10);

   else if(i==10000)

     i = i + (2*10*10*10);
```

```
else if(i==100000)
  i = i + (2*10*10*10*10);



sum = 1;
if(i%10==1)
continue;



int temp = i;
int k = 0 ;



while(temp>0)
{
  int rem = temp%10;
  if(rem==1)
  {
     k = 1;
     break;
   }
  sum = sum*rem;



  temp/=10;
}



if(sum==n && k==0)
{
     c++;

   }

}

printf("%d",c);




return 0;

}
```

19. Number of triangles

```
#include <stdio.h>
int main()
{
 int t;
 long long int n,b1,b2;
 scanf("%d",&t);
 while(t--)
 {
  scanf("%lld %lld %lld",&n,&b1,&b2);
  long double sum=0;
  int z=0;
  if(b1>b2)
   z=b1-b2;
  else
   z=b2-b1;
  int v1=z-1;
  int v2=n-z-1;
  if(v1>0&&v2>0)
   sum=(v1+v2-2)*(n-4);
  else
   sum=(v1+v2-1)*(n-4);
  if(v1>2)
   sum=sum-v1+2;
  if(v2>2)
   sum=sum-v2+2;
  printf("%.Lf\n",sum);
 }
 return 0;
}
```

20. Odd divisors

```
#include<stdio.h>
int main()
{
 int t;
 scanf("%d",&t);
 while(t--)
 {
  long int n,m;
  scanf("%ld%ld",&n,&m);
  long int sum=0;
  while(n>0)
  {
   sum+=((n/2+n%2)%m)*((n/2+n%2)%m);
   sum=sum%m;
   n=n/2;
  }
  printf("%ld\n",sum);
 }
 return 0;
}
```

21. Number of cycles

```
#include <stdio.h>

int main(){
 int t;
 scanf("%d",&t);
 while(t--)
 {
  long long int n;
  scanf("%lld",&n);
  printf("%lld\n",n*(n-1) + 1);
 }
}
```

22. Rain sound

```
#include<stdio.h>

int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int l,r,s;
        scanf("%d%d%d",&l,&r,&s);
        int min,max;
        min=l/s;
        if(min*s < l)
            min++;

        max=r/s;
        if(min<=max)
            printf("%d %d\n",min,max);
        else
            printf("-1 -1\n");
    }
}
```

23. Interest degree of vertices

```
#include<stdio.h>
const int maxn = 1e4 + 17, mod = 1e9 + 7;

int n;

int main(){

    scanf("%d",&n);

    int pa = 1;

    for(int i = 1; i < n; i++){

        int p;

        scanf("%d",&p);

        pa &= p == i;

    }

    long long sum = 0;

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

        int x;

        scanf("%d",&x);

        sum += x;

    }

    printf("%lld\n", !pa * sum );

}
```

24. Distribute chocolates

```
#include<stdio.h>
int main()

{

    int t;

    scanf("%d",&t);

    while(t--)

    {
        unsigned long long int c , n,given;

        scanf("%llu%llu",&c,&n);

given = (n*(1+n))/2;

     if(given>c)

     {

         printf("%llu\n",c);

     }

     else

     {

        c = c - given ;

     printf("%llu\n",c%n);

}
    }
    return 0 ;

}
```

25. Case conversion

```
#include<stdio.h>
#include<stdbool.h>
#include<malloc.h>
#include<ctype.h>

char s[105];
char temp[200];
char* caseConversion (char *s) {
   int k=0;
   for(int i=0;s[i]!='\0';i++)
   {
       if(i==0)
       {
           if(isupper(s[i]))
           temp[k]=tolower(s[i]);
           else
           temp[k]=s[i];
           //printf("%c",temp[k]);
            k++;
       }
       else
       {
           if(isupper(s[i]))
           {
               temp[k]='_';
```

```
          //printf("%c",temp[k]);
           k++;
          temp[k]=tolower(s[i]);
         // printf("%c",temp[k]);
           k++;
       }
       else
       {
          temp[k]=s[i];
          //printf("%c",temp[k]);
           k++;
       }
     }

  }
  temp[k]='\0';
  //printf("%s\n",temp);
  return temp;
}

int main() {
   int T;
   scanf("%d", &T);
   for(int t_i=0; t_i<T; t_i++)
   {
      scanf("%s", s);

      char* out_ = caseConversion(s);
      printf("%s", out_);
      printf("\n");
   }
}
```

26. Teachers and students

```
#include<stdio.h>
#include<stdbool.h>
#include<malloc.h>

long long int fun (int N ) {
  // Your code goes here
  //int n= sizeof(A);

  //printf("%lld",k);
  //printf("N%d  \n",N);
  long long int k= pow(2, N*2);
  return k;
}

int main() {
   int T;
   scanf("%d", &T);
   for(int t_i=0; t_i<T; t_i++)
   {
      int N;
```

```
        scanf("%d", &N);

        long long int K;
        scanf("%lld", &K);

        for(int i=0; i<N; i++){
            scanf("%lld", &K);
        }

        long long int k= 1;
        N*=2;
        while(N>0){
            k*=2;
            k=k% 1000000007;
            N--;
        }

       // long long int out_ = fun(N);
        printf("%lld", k);
        printf("\n");
    }
}
```

27. Color the boxes

```
#include <stdio.h>
 int main(){
 unsigned long long int N, M, ways, fact;
 scanf("%llu %llu", &N , &M);
 fact= 1;
 while(M>0){
  fact = fact*M;
  fact= fact%1000000007;
  M-=1;
 }
 printf("%llu", fact);

}
```

28. Moving people

```
#include <stdio.h>

    static int parseNum() {
      int c, n;
      int neg = 1;
      n = getchar_unlocked();

      if (n == '-') {
        neg = -1;
        n = getchar_unlocked() - '0';
      } else {
        n = n - '0';
      }
      while ((c = getchar_unlocked()) >= '0')
        n = 10 * n + c - '0';
      n *= neg;
      return n;
    }
    int grid[1000][1000];

    void removeCol(int actualX, int* total, int row, int col) {
      if (actualX > 0) {
        actualX -=1;
      } else {
        actualX += col;
      }
      if (actualX < 0 || actualX >= col) {
        return;
      }
      for (int y = 0; y < row; y++) {
        (*total) -= grid[actualX][y];
        grid[actualX][y] = 0;
      }
    }
    void removeRow(int actualY, int* total, int row, int col) {
      if (actualY > 0) {
        actualY -=1;
      } else {
        actualY += row;
      }
      if (actualY < 0 || actualY >= row) {
        return;
      }
      for (int x = 0; x < col; x++) {
        (*total) -= grid[x][actualY];
        grid[x][actualY] = 0;
      }
    }
    int to = 0;

    int main() {
      int n, m, q;
      int actualX = 0, actualY = 0;
      int total = 0;
      int xMin = 0;
      int yMin = 0;
      int xMax = 0;
      int yMax = 0;

      n = parseNum();
      m = parseNum();
      q = parseNum();
      // printf("%d %d %d\n", n,m,q);
      // xMax = m - 1;
      // yMax = n - 1;

      for (int y = 0; y < n; y++) {
        for (int x = 0; x < m; x++) {
          grid[x][y] = getchar_unlocked() - '0';
```

```
        if (grid[x][y] == 1) {
          total++;
        }
      }
      getchar_unlocked();
    }

    for (int qc = 0; qc < q; qc++) {
      int instruct = parseNum();
      // printf("instruct: %d\n", instruct);
      if (instruct == 1) {
        int tempX = parseNum();
        int tempY = parseNum();
        // printf("%d,%d\n", tempX, tempY);

        actualX += tempX;
        actualY += tempY;

        if (actualX < xMax) {
          for (int r = actualX; r < xMax; r++) {
            removeCol(r, &total, n, m);
          }
          xMax = actualX;
        }

        if (actualY < yMax) {
          for (int r = actualY; r < yMax; r++) {
            removeRow(r, &total, n, m);
          }
          yMax = actualY;
        }

        if (actualX > xMin) {
          for (int r = actualX; r > xMin; r--) {
            removeCol(r, &total, n, m);
          }
          xMin = actualX;
        }

        if (actualY > yMin) {
          for (int r = actualY; r > yMin; r--) {
            removeRow(r, &total, n, m);
          }
          yMin = actualY;
        }
      } else if (instruct == 2) {
        printf("%d\n", total);
      }
      // getchar_unlocked();
    }

    return 0;
  }
```

29. Super balanced bracket

```
#include<stdio.h>
#include<string.h>
int main(){
 int t;
 scanf("%d",&t);
 for(int i=0;i<t;i++){
  char s[100000];
  scanf("%s",&s);
  int count=0;
  for(int i=0;i<strlen(s)/2;i++){
   if(s[i]=='(')
   count++;
  }
   printf("%d\n",2*count);
 }
}
```

30. Tic-tac-toe

```
#include<stdio.h>
int main(){
 char t[3][3];
 int x=0,o=0,r=0,f=0;
 for(int i=0;i<3;i++){
  scanf("%s",t[i]);
 }
 for(int i=0;i<3;i++){
  for(int j=0;j<3;j++){
   if(t[i][j]=='X')
   x+=1;
   else if(t[i][j]=='O')
   o+=1;
   else
   r++;
  }
 }
 if((o && x==0) || x-o>1 || o-x>1)
 printf("Wait, what?");
 else{
  if((t[0][0]=='X' && t[0][1]=='X' && t[0][2]=='X')||(t[1][0]=='X' && t[1][1]=='X' && t[1][2]=='X')||(t[2][0]=='X' && t[2][1]=='X' && t[2][2]=='X')||(t[0][0]=='X' && t[1][0]=='X' && t[2][0]=='X')||(t[0][1]=='X' && t[1][1]=='X' && t[2][1]=='X')||(t[0][2]=='X' && t[1][2]=='X' && t[2][2]=='X')||(t[0][0]=='X' && t[1][1]=='X' && t[2][2]=='X')||(t[0][2]=='X' && t[1][1]=='X' && t[2][0]=='X')){
   if(x-o==1)
   printf("X won.");
   else
   printf("Wait, what?");
  }
  else if((t[0][0]=='O' && t[0][1]=='O' && t[0][2]=='O')||(t[1][0]=='O' && t[1][1]=='O' && t[1][2]=='O')||(t[2][0]=='O' && t[2][1]=='O' && t[2][2]=='O')||(t[0][0]=='O' && t[1][0]=='O' && t[2][0]=='O')||(t[0][1]=='O' && t[1][1]=='O' && t[2][1]=='O')||(t[0][2]=='O' && t[1][2]=='O' && t[2][2]=='O')||(t[0][0]=='O' && t[1][1]=='O' && t[2][2]=='O')||(t[0][2]=='O' && t[1][1]=='O' && t[2][0]=='O')){
   if(o-x==0)
   printf("O won.");
   else
   printf("Wait, what?");
  }
  else if(r==0)
  printf("It's a draw.");
  else if(x==o)
  printf("X's turn.");
  else
  printf("O's turn.");
 }
}
```

31. Deleting Numbers

```
#include<stdio.h>
int min = 0;
int MaxFind(int n, int k, int a[n], int flag)
{
  if(flag == 1){k -= 1;}
  for (int changer = ((n / 2) - (k / 2)); changer < ((n / 2) + (k / 2) + 1); changer++)
  {
    if(min <= a[changer]){min = a[changer];}
  }
  return min;
}
int main()
{
  int n = 0, k = 0;
  scanf("%d %d", &n, &k);
  int a[n];
  for (int i = 0; i < n; i++)
  {
    scanf("%d", &a[i]);
  }
  if(n == 3 && k == 1)
  {
    for (int changer = ((n / 2) - (k / 2)); changer < ((n / 2) + (k / 2) + 1); changer++)
    {
      if(min <= a[changer]){min = changer;}
    }
    if(a[min - 1] >= a[min]){printf("%d\n",a[min - 1]); return 0;}
    printf("%d\n",a[min]);
  }
  if((n % 2 != 0 && k % 2 == 0) || (n % 2 == 0 && k % 2 != 0)){MaxFind(n, k, a, 0);}
  else{MaxFind(n, k, a, 1);}
  printf("%d \n",min);
}
```

32. An equilateral triangle

```
#include<stdio.h>
int countOdd(int n)
{
 int coun = 0, m, j, i;
 for (i = n - 2; i >= 1; i--)
 {
  if (i & 1)
  {
   m = (n - i) / 2;
   j = (i * (i + 1)) / 2;
   coun += j * m;
  }
  else
  {
   m = ((n - 1) - i) / 2;
   j = (i * (i + 1)) / 2;

   coun += j * m;
  }
 }
 return coun;
}

int countEven(int n)
{
 int coun = 0, m, j, i;
 for (i = n - 2; i >= 1; i--)
 {
  if (i & 1)
  {
   m = ((n - 1) - i) / 2;
   j = (i * (i + 1)) / 2;
   coun += j * m;
  }
  else
  {
   m = (n - i) / 2;
   j = (i * (i + 1)) / 2;
   coun += j * m;
  }
 }
 return coun;
}

int main()
{
 int n;
 scanf("%d",&n);
 if (n & 1)
  // cout << countOdd(n);
  printf("%d",countOdd(n));
 else
  // cout << countEven(n);
  printf("%d",countEven(n));
 return 0;
}
```