سوال اول

لیست کلاسی

علی تازه کلاس اول را تمام کرده و در این کلاس با حروف الفبا و همجنین با شمارش اعداد اشنا شده. یک روز در مدرسه چشمش به لیستی از اسامی همکلاسی‌هایش افتاد و از روی کنجکاوی, تصمیم گرفت که ببیند اسم کدام یک از دوست هایش از تعداد حروف متفاوت بیشتری تشکیل شده. از آنجا که هنوز مهارت کافی برای این کار را ندارد از شما خواسته که به او در انجام این کار کمک کنید.

ورودی:

خط اول ورودی شامل عدد n است. در n خط بعدی هر خط شامل یک اسم است. هر اسم رشته‌ای با حداکثر ۲۰ حرف از حروف کوچک انگلیسی می‌باشد.

$$1 \leq n \leq 100$$

خروجی:

در تنها خط خروجی یک عدد چاپ کنید که برابر تعداد بیشترین حروف مختلف در بین اسم‌های لیسن خواهد بود.

ورودی نمونه:

```
6
alireza
parsa
reza
sepehr
mohammad
karim
```

خروجی نمونه:

```
6
```

راه حل:

```c
#include <stdio.h>
int diff_letters[100];
int main()
{
    char esm[21];
    int n , javab = 0;
    scanf("%d\n", &n);
    for (int k = 0; k < n; k++)
    {
        scanf("%s\n", esm);
        for (int i = 0; i < 22; i++)
```

سوال اول

لیست کلاسی

علی تازه کلاس اول را تمام کرده و در این کلاس با حروف الفبا و همجنین با شمارش اعداد اشنا شده. یک روز در مدرسه چشمش به لیستی از اسامی همکلاسی‌هایش افتاد و از روی کنجکاوی, تصمیم گرفت که ببیند اسم کدام یک از دوست هایش از تعداد حروف متفاوت بیشتری تشکیل شده. از آنجا که هنوز مهارت کافی برای این کار را ندارد از شما خواسته که به او در انجام این کار کمک کنید.

ورودی:

خط اول ورودی شامل عدد n است. در n خط بعدی هر خط شامل یک اسم است. هر اسم رشته‌ای با حداکثر ۲۰ حرف از حروف کوچک انگلیسی می‌باشد.

$$1 \leq n \leq 100$$

خروجی:

در تنها خط خروجی یک عدد چاپ کنید که برابر تعداد بیشترین حروف مختلف در بین اسم‌های لیسن خواهد بود.

ورودی نمونه:

```
6
alireza
parsa
reza
sepehr
mohammad
karim
```

خروجی نمونه:

```
6
```

راه حل:

```c
#include <stdio.h>
int diff_letters[100];
int main()
{
    char esm[21];
    int n , javab = 0;
    scanf("%d\n", &n);
    for (int k = 0; k < n; k++)
    {
        scanf("%s\n", esm);
        for (int i = 0; i < 22; i++)
        {
           if (esm[i] < 'a' || esm[i] >'z')
                break;
            for (int j = 0; j < i; j++)
            {

                if ( esm[i] != esm[j])
                    continue;
                else
                {
                    diff_letters[k] -= 1;
                    break;
                }

            }
            diff_letters[k] += 1;
        }

    }
    for (int i = 0; i < n; i++)
    {
        if (diff_letters[i] > javab )
            javab = diff_letters[i];
    }
    printf("%d", javab);

}
```

سوال دوم

فیبوناچی گنده

- محدودیت زمانی :۱ ثانیه

برادر علی کلاس دهم است و تازه شروع به خواندن المپیاد ریاضی کرده. او اخیرا با مبحث توابع بازگشتی اشنا شده و بسیار از ان لذت برده. این لذت به حدی رسیده که باعث شده او بخواهد باقی مانده جمع اعداد فیبوناچی متوالی را به پیمانه یک عدد دلخواه خروجی دهد. حال چون او درگیر درس های دیگر المپیادش (هندسه, جبر و نظریه اعداد) است, او از شما میخواهد تا این کار را برایش انجام دهید.

ورودی:

خط اول ورودی شامل سه عدد n, m, k است.

$$0 < k < 1000 \quad , \quad 0 \leq n, m < 10^{13}$$

خروجی:

شما باید جمع nامین عدد فیبوناچی تا mامین عدد فیبوناچی را حساب کنید و حاصل را به پیمانه k خروجی دهید.

ورودی نمونه:

```
0 9999999999999 2
```

خروجی نمونه:

```
0
```

ورودی نمونه:

```
4 7 10
```

خروجی نمونه:

```
9
```

دقت کنید صفر اُمین عدد فیبوناچی صفر, اولین عدد فیبوناچی ۱ در نظر گرفته شده.

راه حل:

```
#include <stdio.h>

long long period(long long m){
    long long previous = 0;
    long long current = 1;
    int i = 0;
    while(1){
        long long temp = previous % m;
        previous = current % m;
        current = (current +temp) % m;
        i++;
        if (previous == 0 && current == 1)
            break;
    }
    return i;
}

long long getFibonacciPartialSumNaive(long long n, long long m) {
       if (n < 0) return 0;
       if (n <= 1) return n;
       long long previous = 0;
       long long current  = 1;
       long long sum = 1;
       for (long long i = 0; i < n % period(m) - 1; i++) {
           long long tmp_previous = previous % m;
           previous = current % m;
           current = (tmp_previous + current) % m;
           sum = (sum + current) % m;
       }
       if (n % period(m) == 0)
           return 0;
       return sum;
   }

int main() {
    long long from;
    long long to;
    long long k;
    scanf("%lld %lld %lld", &from, &to, &k);
                long    long    ans    =    getFibonacciPartialSumNaive(to,    k)    -
getFibonacciPartialSumNaive(from - 1,k);
    if (to == 0){
        printf(0);
        return 0;
    }
    if (ans < 0)
        ans += k;
    printf("%lld", ans);
}
```

سوال اول

لیست کلاسی

علی تازه کلاس اول را تمام کرده و در این کلاس با حروف الفبا و همجنین با شمارش اعداد اشنا شده. یک روز در مدرسه چشمش به لیستی از اسامی همکلاسی‌هایش افتاد و از روی کنجکاوی, تصمیم گرفت که ببیند اسم کدام یک از دوست هایش از تعداد حروف متفاوت بیشتری تشکیل شده. از آنجا که هنوز مهارت کافی برای این کار را ندارد از شما خواسته که به او در انجام این کار کمک کنید.

ورودی:

خط اول ورودی شامل عدد n است. در n خط بعدی هر خط شامل یک اسم است. هر اسم رشته‌ای با حداکثر ۲۰ حرف از حروف کوچک انگلیسی می‌باشد.

$$1 \leq n \leq 100$$

خروجی:

در تنها خط خروجی یک عدد چاپ کنید که برابر تعداد بیشترین حروف مختلف در بین اسم‌های لیسن خواهد بود.

ورودی نمونه:

```
6
alireza
parsa
reza
sepehr
mohammad
karim
```

خروجی نمونه:

```
6
```

راه حل:

```c
#include <stdio.h>
int diff_letters[100];
int main()
{
    char esm[21];
    int n , javab = 0;
    scanf("%d\n", &n);
    for (int k = 0; k < n; k++)
    {
        scanf("%s\n", esm);
        for (int i = 0; i < 22; i++)
        {
           if (esm[i] < 'a' || esm[i] >'z')
                break;
            for (int j = 0; j < i; j++)
            {

                if ( esm[i] != esm[j])
                    continue;
                else
                {
                    diff_letters[k] -= 1;
                    break;
                }

            }
            diff_letters[k] += 1;
        }

    }
    for (int i = 0; i < n; i++)
    {
        if (diff_letters[i] > javab )
            javab = diff_letters[i];
    }
    printf("%d", javab);

}
```

سوال دوم

فیبوناچی گنده

- محدودیت زمانی :۱ ثانیه

برادر علی کلاس دهم است و تازه شروع به خواندن المپیاد ریاضی کرده. او اخیرا با مبحث توابع بازگشتی اشنا شده و بسیار از ان لذت برده. این لذت به حدی رسیده که باعث شده او بخواهد باقی مانده جمع اعداد فیبوناچی متوالی را به پیمانه یک عدد دلخواه خروجی دهد. حال چون او درگیر درس های دیگر المپیادش (هندسه, جبر و نظریه اعداد) است, او از شما میخواهد تا این کار را برایش انجام دهید.

ورودی:

خط اول ورودی شامل سه عدد n, m, k است.

$$0 < k < 1000 \quad , \quad 0 \leq n, m < 10^{13}$$

خروجی:

شما باید جمع nامین عدد فیبوناچی تا mامین عدد فیبوناچی را حساب کنید و حاصل را به پیمانه k خروجی دهید.

ورودی نمونه:

```
0 9999999999999 2
```

خروجی نمونه:

```
0
```

ورودی نمونه:

```
4 7 10
```

خروجی نمونه:

```
9
```

دقت کنید صفر اُمین عدد فیبوناچی صفر, اولین عدد فیبوناچی ۱ در نظر گرفته شده.

راه حل:

```
#include <stdio.h>
long long array[1000];
long long fib(int n, long long array[]) {
    if(n == 1) {
        if(array[n] == 0)
            array[n] = 1;
        return array[n];
    }
    if(n == 2) {
        if(array[n] == 0)
            array[n] = 1;
        return array[n];
    }
    if(array[n] == 0)
        array[n] = fib(n - 1, array) + fib(n - 2, array);
    return array[n];
}

long long period(long long m){
    long long previous = 0;
    long long current = 1;
    int i = 0;
    while(1){
        long long temp = previous % m;
        previous = current % m;
        current = (current +temp) % m;
        i++;
        if (previous == 0 && current == 1){
            break;
        }
    }
    return i;
}

long long getFibonacciPartialSumNaive(long long n, long long m) {
       if (n < 0){
           return 0;
       }
       if (n <= 1)
           return n;

       long long sum = 0;
       for (long long i = 0; i <= n % period(m) - 1; i++) {
           sum = (sum + array[i + 1]) % m;
       }
       if (n % period(m) == 0){
           return 0;
       }
       return sum;
   }
```

```
int main() {
    long long from;
    long long to;
    long long k;
    fib(1000, array);
    scanf("%lld %lld %lld", &from, &to, &k);
          long  long  ans  =  getFibonacciPartialSumNaive(to,  k)  -
getFibonacciPartialSumNaive(from - 1,k);
    if (to == 0){
        printf(0);
        return 0;
    }
    if (ans < 0)
        ans += k;
    printf("%lld", ans);
}
```