

SPOJ MLLOVE – Masud and Layla love story

Nama : Timothy Hosia

NRP : 5025211098

Permasalahan dan pendekatan :

Soal ini pertama meminta user untuk memasukkan 2 input nilai yang diberi nama n dan k . Kemudian akan timbul sebuah deret baris dengan $(1+ik)$ th index.. ($i = 0,1,2,3,4,5... \text{Infinity}$) \rightarrow dalam kata lain merupakan sebuah deret dengan a adalah 1 dan dengan b adalah k , kemudian deret baris itu di modulo dengan n . hasil modulo itu disimpan dalam index berurut. Angka dari hasil tersebut akan dijumlahkan ke hasil asalkan angka tersebut belum pernah dimasukan ke dalam index (angka yang dijumlahkan selalu baru).

Example:

Let's say the number $n = 6$ and $k = 3$

Now Layla can go to $(1+ik)$ th index.. ($i = 0,1,2,3,4,5... \text{Infinity}$)

that's mean she will go index = 1 , 4 , 7 , 10 , 13 , 16 ... Infinity.

then the cost of

1st index price is $(1\%6) = 1$.

2nd index price is $(4\%6) = 4$.

3rd index price is $(7\%6) = 1$.

4th index price is $(10\%6) = 4$.

and it will continue.

At the end of the process, she will collect 1 and 4 cost prices gift. so the sum of all distinct price is 5.

Seperti yang telah dijelaskan di contoh test case soal, nilai n dan k yaitu 6 dan 3, akan menghasilkan deret 1, 4, 7, 10, 13, dst. Kemudian deret tersebut akan

dimodulo dengan 6. Urutan pertama diisi dengan nilai 1, kemudian 4, kemudian 1 kembali dan terus berulang. Jadi total angka yang berbeda yang dihasilkan adalah 1 ditambahkan 4 sehingga menghasilkan output 5.

Untuk mencari solusi, kita akan melihat setiap test case untuk melihat pola yang mungkin berulang.

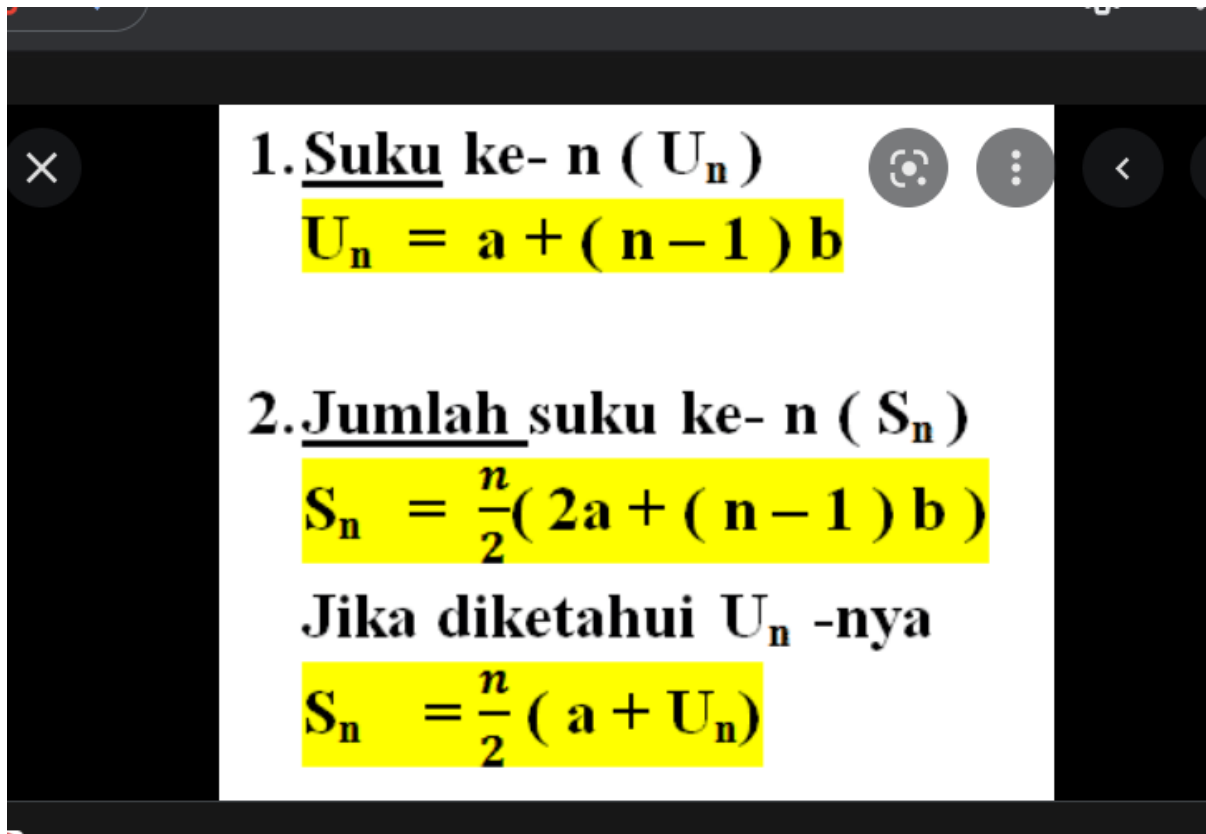
	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		1	2	3	4	5	6	7	8	9	10	K
3	1	0	0	0	0	0	0	0	0	0	0	0
4	2	1	1	1	1	1	1	1	1	1	1	1
5	3	3	3	1	3	3	0	3	3	0	3	
6	4	6	4		1							
7	5	10	10	10	10	1	10	10	10	10	0	
8	6	15	9			1						
9	7	21	21	21	21	21	1	21	21	21		
10	8	28	16					1				
11	9	36	36	36	36	36	36	36	1	36		
12	10										1	
13	N											

Dari percobaan pola pertama ditemukan bahwa bila $n = 1$ akan menghasilkan nilai 0. Kemudian jika nilai $n = 2$ akan menghasilkan nilai 1. Temuan lainnya adalah jika n memiliki nilai yang sama dengan k akan menghasilkan nilai 1. Kemudian di nilai n ganjil (lebih dari 1) akan menghasilkan nilai yang sama yaitu penjumlahan semua nilai positif sebelum angka itu. Di pembahasan selanjutnya akan ditemukan bahwa hasil tersebut diambil dari rumus **deret matematika**.

Belum puas dengan hasilnya, akhirnya saya mencoba melihat pola lainya dengan mencoba menyusun hasil test case dengan susunan yang berbeda.

16	$n = 7$	$k = 3$			$n = 6$	$k = 5$	
17	Index	modulo	Hasil		Index	modulo	Hasil
18	1	7	1		1	6	1
19	4	7	4		6	6	0
20	7	7	0		11	6	5
21	10	7	3		16	6	4
22	13	7	6		21	6	3
23	16	7	2		26	6	2
24	19	7	5		31	6	1
25	22	7	1		36	6	0
26	25	7	4		41	6	5
27							
28							

Setelah disusun seperti berikut dan dibandingkan dengan pola pertama yang saya temukan. Saya menemukan bahwa hasil tersebut dapat ditemukan dari sebuah deret bilangan. Di bilangan n ganjil (lebih dari 1) akan menghasilkan nilai yang sama yaitu penjumlahan semua nilai positif sebelum angka itu. Setelah dilihat dari mana deret angka tersebut bisa diambil, saya menemukan regularity yang sesuai dengan rumus **deret aritmatika**.



1. Suku ke- n (U_n)

$$U_n = a + (n - 1) b$$

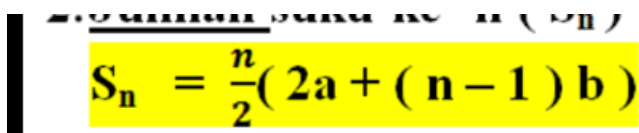
2. Jumlah suku ke- n (S_n)

$$S_n = \frac{n}{2} (2a + (n - 1) b)$$

Jika diketahui U_n -nya

$$S_n = \frac{n}{2} (a + U_n)$$

Kemudian saya mencoba menerapkan rumus yang ada sesuai dengan pola bilangan yang ada. Saya menemukan bahwa



2. Jumlah suku ke- n (S_n)

$$S_n = \frac{n}{2} (2a + (n - 1) b)$$

Rumus ini sesuai dengan mencari hasil selain nilai nilai special sebelumnya. Kemudian saya merumuskan kode tersebut menjadi source code.

*beberapa kali saya mencoba merumuskan source code tanpa fungsi (dapat dilihat di comment yang ada di sc jawaban saya) untuk mencoba beberapa algoritma yang dapat digunakan. Kemudian Ketika saya ingin mengimpementasikan rumus deret aritmatika ini, saya mulai

mengimplementasikan penggunaan fungsi dan metode read karya Bapak untuk mempersingkat waktu. Berikut adalah kode yang dapat saya rumuskan.

```
#include <stdio>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <algorithm>

using namespace std;

#define gc getchar
#define ULL unsigned long long

ULL read(){
    ULL value = 0; bool ne=0;
    char c = gc();
    while(c==' ' or c=='\n') c =gc();
    if(c=='-'){ne = 1;c = gc();}
    while(c>='0' and c<='9'){
        value = (value<<3)+(value<<1)+c-'0';c = gc();}
    if(ne) value*=-1;
    return value;
}

//ULL sn(ULL a,ULL n,ULL b){
//
//}

ULL algo(ULL n, ULL k){
    ULL jml, result = 0, fpb, a;
    if(n==1)
    {
        result = 0;
    }
    else if(n==2)
    {
        result = 1;
    }
    else if(n==k)
    {
        result = 1;
    }
    else
    {
        ULL fpb = __gcd(n,k);
        printf("fpb adalah %d", fpb);
        if(fpb>1)
```

```

    {
        a=1;
        result = ((n / fpb) / 2) * (2 + ((n / fpb) - 1) * fpb);
    }
    else if(fpb==1)
    {
        int a=0;
        result = (n * (n - 1)) / 2;
    }
}
return result;
}

void selesaikan(){
    ULL n = read(); ULL k = read();
    ULL hasil = algo(n, k);
    printf("%lld", hasil);
}

int main(){
    selesaikan();
    return 0;
}

```

Dimulai dari line 1-5 adalah library yang pada umumnya saya gunakan, ditambahkan library algorithm untuk mencari fpb menggunakan __gcd

```

    }
    else
    {
        ULL fpb = __gcd(n, k);
        printf("fpb adalah %d", fpb);
    }
}

```

Kemudian kita menggunakan define gc yaitu getchar dan ULL untuk melambangkan unsigned long long.

Kemudian kita melihat line 59-68 untuk melihat fungsi main dan “selesaikan” yang dipanggil untuk menyelesaikan soal dengan memanggil fungsi

menggunakan

passing

parameter.

```
58
59 void selesaikan(){
60     ULL n = read(); ULL k = read();
61     ULL hasil = algo(n, k);
62     printf("%lld", hasil);
63 }
64
65 int main(){
66     selesaikan();
67     return 0;
68 }
69
```

Dilihat di main memanggil fungsi selesaikan, dan fungsi selesaikan akan mengassign nilai n dan k dengan meminta input user menggunakan fungsi read karya bapak.

```
1
2 ULL read(){
3     ULL value = 0; bool ne=0;
4     char c = gc();
5     while(c==' ' or c=='\n') c = gc();
6     if(c=='-'){ne = 1; c = gc();}
7     while(c>='0' and c<='9'){
8         value = (value<<3)+(value<<1)+c-'0'; c = gc();}
9     if(ne) value*=-1;
0     return value;
1 }
2
```

Kemudian kita meng assign ull hasil dengan fungsi “algo” kemudian meng outputkan nilai hasil tersebut.

```
ULL algo(ULL n, ULL k){
    ULL jml, result = 0, fpb, a;
    if(n==1)
    {
        result = 0;
    }
    else if(n==2)
    {
        result = 1;
    }
    else if(n==k)
    {
        result = 1;
    }
}
```

```

else
{
    ULL fpb = __gcd(n,k);
    // printf("fpb adalah %d", fpb);
    if(fpb>1)
    {
        a=1;
        result = ((n / fpb) / 2) * (2 + ((n / fpb) - 1) * fpb);
    }
    else if(fpb==1)
    {
        int a=0;
        result = (n * (n - 1)) / 2;
    }
}
return result;
}

```

Kemudian kita berada diinti fungsi penyelesaian.

```

LL algo(ULL n, ULL k){
    ULL jml, result = 0, fpb, a;
    if(n==1)
    {
        result = 0;
    }
    else if(n==2)
    {
        result = 1;
    }
    else if(n==k)
    {
        result = 1;
    }
}

```

fungsi menerima passingan nilai n dan k. kemudian kita meng inisiasi variable jml result fpb dan a. Pertama akan menggunakan if else sesuai analisis regularitas yang kita dapat sebelumnya., bila n = 1 hasilnya 0, jika n = 2 hasilnya 1, kemudian jika n= k hasilnya adalah 1.

```

    }
    else
    {
        ULL fpb = __gcd(n,k);
        // printf("fpb adalah %d", fpb);
        if(fpb>1)
        {
            a=1;
            result = ((n / fpb) / 2) * (2 + ((n / fpb) - 1) * fpb);
        }
        else if(fpb==1)
        {
            int a=0;
            result = (n * (n - 1)) / 2;
        }
    }
    return result;
}

```

Pertama kita meng assign variable fpb dengan hasil pencarian fpb dengan gcd.

Kemudian akan dipecah menjadi 2 kasus. Jika fpb 1, nilai a akan di assign nilai 1 kemudian menggunakan rumus yang telah dibahas sebelumnya. Kemudian jika fpb nya hanya 1 nilai a akan di sama dengankan 0, kemudian menggunakan rumus diatas akan meng assign nilai result. Kemudian fungsi akan me return result yang kemudian akan di cetak.

PSEUDOCODE

```

mendeklarasi gc getchar
mendeklarasi ULL unsigned Long Long

algoritma sakkti karya bapak rully ULL read(){
    ULL value = 0; bool ne=0;
    char c = gc();
    while(c==' ' or c=='\n') c =gc();
    if(c=='-'){ne = 1;c = gc();}
    while(c>='0' and c<='9'){
        value = (value<<3)+(value<<1)+c-'0';c = gc();}
    if(ne) value*=-1;
    return value;
}

```


Algoritma ULL algo(ULL n, ULL k)// passing parameter n dan k

```
Inisialisasi variable yang akan digunakan ULL jml, result = 0, fpb, a;
Jika (n==1)
    result -> 0;

else if(n==2)
    result -> 1;

else if(n==k)
    result -> 1;

else
    assign variable ULL fpb = __gcd(n,k);

    if(fpb>1)
        nilai a=1;
        nilai result = ((n / fpb) / 2) * (2 + ((n / fpb) - 1) * fpb);

    else if(fpb==1)
        inisialisasi int a=0;
        nilai result = (n * (n - 1)) / 2;

return nilai result;
```

void selesaikan

```
inisialisasi ULL n = meminta input user dengan read ; ULL k = meminta input
user dengan read;
```

```
kemudian assign variable ULL hasil = hasil dari algo(n, k);
cetak "%lld" hasil
```

```
int main
    panggil fungsi selesaikan();
    return 0;
```