

Print n, n-k, n-2k, n-3k.... till l

Problem

Submissions

Leaderboard

Discussions

You will be given three integer inputs **N**, **K** and **L** and you to print the series **N**, **N-K**, **N-2K**, **N-3K**, ... till last where the value printed in the end should be just greater than or equal to the given input **L**.

To be clear: You will print **L** if **L** belongs to the series.

Input Format

For each test case, you will get

N in the first line as an integer input,

K in the second line as an integer input,

L in the third line as an integer input.

Constraints

$0 \leq N, K < 2^{10}$

$-2^{31} \leq L \leq 2^{31}-1$

Output Format

You have to print the series where each number should be printed in a separate line.

Sample Input 0

```
50
5
4
```

Sample Output 0

Submitted Code

Language: Java 15

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n= sc.nextInt();
9         int k=sc.nextInt();
10        int l=sc.nextInt();
11
12        for(int i = n ;i>=l;i-=k)System.out.println(i);
13    }
14 }
```

Sample Output 0

```
50
45
40
35
30
25
20
15
10
5
```

Sample Input 1

```
70
9
10
```

Sample Output 1

```
70
61
52
43
34
25
16
```

$i = n$
 $50 - 5 = 45$
 $45 - 5 = 40$
 $40 - 5 = 35$

$50 - 5 = 45$
 $45 - 5 = 40$
 $40 - 5 = 35$

$n = 50$
 $n - 1 \times k = 45$
 $50 - 1 \times 5 = 45$
 $n - 2 \times k = 40$
 $50 - 2 \times 5 = 40$
 $50 - 3 \times 5 = 35$

$n = 50$
 $n - 1 \times k = 45$
 $50 - 1 \times 5 = 45$
 $n - 2 \times k = 40$
 $50 - 2 \times 5 = 40$
 $n - 3 \times k = 35$
 $50 - 3 \times 5 = 35$

The task was given to a novice programmer named Kim to print the first **n** numbers of the series 5, 11, 17, 23, 29, 35, 41 using a **while loop**. Kim took the input value of **n** from the user and completed the task successfully.

Input Format

Given a Int Input N

Constraints

NA

Output Format

Print the sequence till N

Sample Input 0

50

Sample Output 0

5
11
17
23
29
35
41
47

Explanation 0

Printed All the Number till 50 of the following sequence

Submitted Code

Language: Java 15

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9         int i = 5;
10        while(i<=n){
11            System.out.println(i);
12            i+=6;
13        }
14    }
15 }
```

Handwritten diagram illustrating the sequence generation process. It shows the sequence 5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 65, 71, 77, 83, 89, 95, 101, 107, 113, 119, 125, 131, 137, 143, 149, 155, 161, 167, 173, 179, 185, 191, 197, 203, 209, 215, 221, 227, 233, 239, 245, 251, 257, 263, 269, 275, 281, 287, 293, 299, 305, 311, 317, 323, 329, 335, 341, 347, 353, 359, 365, 371, 377, 383, 389, 395, 401, 407, 413, 419, 425, 431, 437, 443, 449, 455, 461, 467, 473, 479, 485, 491, 497, 503, 509, 515, 521, 527, 533, 539, 545, 551, 557, 563, 569, 575, 581, 587, 593, 599, 605, 611, 617, 623, 629, 635, 641, 647, 653, 659, 665, 671, 677, 683, 689, 695, 701, 707, 713, 719, 725, 731, 737, 743, 749, 755, 761, 767, 773, 779, 785, 791, 797, 803, 809, 815, 821, 827, 833, 839, 845, 851, 857, 863, 869, 875, 881, 887, 893, 899, 905, 911, 917, 923, 929, 935, 941, 947, 953, 959, 965, 971, 977, 983, 989, 995, 1001. Brackets indicate the addition of 6 to the previous number to get the next number in the sequence.

Handwritten diagram illustrating the sequence generation process. It shows the sequence 5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 65, 71, 77, 83, 89, 95, 101, 107, 113, 119, 125, 131, 137, 143, 149, 155, 161, 167, 173, 179, 185, 191, 197, 203, 209, 215, 221, 227, 233, 239, 245, 251, 257, 263, 269, 275, 281, 287, 293, 299, 305, 311, 317, 323, 329, 335, 341, 347, 353, 359, 365, 371, 377, 383, 389, 395, 401, 407, 413, 419, 425, 431, 437, 443, 449, 455, 461, 467, 473, 479, 485, 491, 497, 503, 509, 515, 521, 527, 533, 539, 545, 551, 557, 563, 569, 575, 581, 587, 593, 599, 605, 611, 617, 623, 629, 635, 641, 647, 653, 659, 665, 671, 677, 683, 689, 695, 701, 707, 713, 719, 725, 731, 737, 743, 749, 755, 761, 767, 773, 779, 785, 791, 797, 803, 809, 815, 821, 827, 833, 839, 845, 851, 857, 863, 869, 875, 881, 887, 893, 899, 905, 911, 917, 923, 929, 935, 941, 947, 953, 959, 965, 971, 977, 983, 989, 995, 1001. Brackets indicate the addition of 6 to the previous number to get the next number in the sequence.

Sarah was tasked with printing numbers from x till y , including both x and y , using a while loop. She took inputs for x and y from the user using a while loop, initialized a variable to x , and used a while loop to print the numbers until the value exceeded y . Sarah successfully completed the task, gained experience in using while loops, and felt more confident in her programming abilities.

Input Format

x and y are given to you as Input

Constraints

$0 \leq x < y \leq 10^4$

Output Format

Print from X to Y (both Included)

Sample Input 0

10
15

Sample Output 0

10
11
12
13
14
15

Explanation 0

x is 10 y is 15 Printing Done from 10 to 15

Submitted Code

Language: Java 15

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int x= sc.nextInt();
9         int y =sc.nextInt();
10
11         while(x<=y){
12             System.out.println(x);
13             x++;
14         }
15     }
16 }
```

x, y

$x = 10$

$y = 15$

$x \rightarrow y$

$10 \rightarrow 15$

10, 11, 12, 13, 14, 15

while

~~while~~ while($x \leq y$) $10 \leq 15$

mat

$x = 10$

$y =$

$x \leq y$

..

A programming task was assigned to a beginner named Mike. He was asked to print numbers from **n** to **3** using a **while loop**, where **n** is taken as input from the user. Mike took the input value of **n** from the user and used a while loop to print the numbers in reverse order until **3**.

Input Format

Int Given as Input 'N'

Constraints

-100 < N < 3

Output Format

Print all the numbers from **N** to **3**

Sample Input 0

-9

Sample Output 0

-9
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2

Explanation 0

Printing from n=-9 to 3

Submitted Code

```
Language: java 15
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9         while(n<3){
10             System.out.println(n);
11             n++;
12         }
13     }
14 }
```

Handwritten notes in red ink illustrating the logic and execution of the code:

- Top right: $n < 3$ with n circled.
- Left side: A vertical list of numbers from -9 to 2, corresponding to the sample output.
- Bottom right: A sequence of numbers: $n = 0$, $0, 1, 2 < 3$, and a list of numbers from -5 to 2 with the last three numbers (1, 2, 3) crossed out.

A programming task was assigned to a novice programmer named Max to print a sequence of numbers n , $n-5$, $n-10$, $n-15$, $n-20$... using a while loop. Max took the input value of n from the user and used a while loop to print the numbers in decreasing order. Within the loop, he used decrement the value of the current number by 5 to print the next number until the value became **zero**. Max successfully completed the task, learned how to use while loop, and gained experience in manipulating values in a loop.

Input Format

Integers N as an input value.

Constraints

$1 \leq n \leq 10^6$

Output Format

A series of $n, n-5, n-10, n-15, n-20$...

Sample Input 0

30

Sample Output 0

30 25 20 15 10 5

Explanation 0

starting with $n = 30$ and printed till n is greater than 0 with difference of 5 between each term. therefore series is 30 25 20 15 10 5

Submitted Code

Language: Java 15

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9         while(n > 0){
10             System.out.print(n+" ");
11             n-=5;
12         }
13     }
14 }
```

while

$n, n-5, n-10, n-15, n-20$

$n=30$

30, 25, 20, 15, 10, 5

~~$n-5=0 < 0$~~

< 0

$while(n > 0) \{$

$System.out.print(n)$

$n-=5;$

$\}$

HW_Find Permutation 1

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Given n and r , find the value of nPr .

```
( formula of npr=n!/((n-r)!) ) ✓
```

Input Format

Take 2 input n and r as integer.

Constraints

```
1 <= n,r <= 10^4
```

Output Format

Print a integer as output.

Sample Input 0

```
5
2
```

Sample Output 0

```
20
```

Explanation 0

Take $n = 5$ and $r = 2$.

Output should be 20 by the formulae mentioned above.

Submitted Code

```
Language: java 15
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9         int r = sc.nextInt();
10
11         int nfact=1;
12         for(int i=n;i>0;i--)nfact*=i;
13
14         int nrfact=1;
15         for(int i=n-r;i>0;i--)nrfact*=i;
16
17         int permutation = nfact/nrfact;
18
19         System.out.print(permutation);
20     }
21 }
```

nPr ✓

Formula $\frac{n!}{(n-r)!}$ ✓

$(nPr = \frac{n!}{(n-r)!})$

$n = 5$
 $r = 2$

$\frac{n!}{(n-r)!} \Rightarrow \frac{5!}{(5-2)!} = \frac{5 \times 4 \times 3 \times 2 \times 1}{3 \times 2 \times 1} = 20$

- ① $n, r \Rightarrow 5 \Rightarrow 5 \times 4 \times 3 \times 2 \times 1 = 120$
- ② $n! \Rightarrow \text{while}(n > 0) \{$
 - ① $nfact \times = n;$
 $n--$
 - ② $\text{for}(int i=n; i>0; i--) \{$
 - $nfact \times = i;$

- ① $n!$
- ② $(n-r)!$
- ③ $\frac{n!}{(n-r)!}$
- ④ 3280 ✓

HW_Print fibonacci series 2

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

You are given an input **n** as an integer input , Write a program to print the **alternate** fibonacci numbers starting from the **first** fibonacci till the **nth** fibonacci numbers accordingly , if **nth** fibonacci number is part of the series or not.

Input Format

For each test case, you will get **n** as an integer input.

Constraints

1 <= n <= 50

Output Format

Print the output in a single line.

Sample Input 0

10

Sample Output 0

0 1 3 8 21

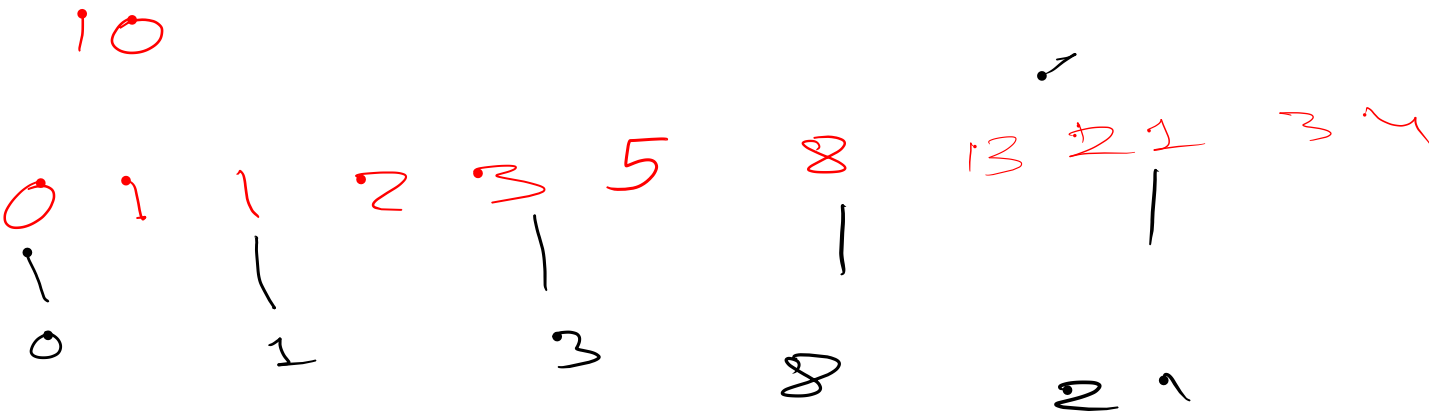
Explanation 0

Alternate fibonacci till **nth** fibonacci are 0 1 3 8 21

Submitted Code

```
Language: Java 15
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n =sc.nextInt();
9         int a=0;
10        int b=1;
11        int sum=0;
12        for(int i=1;i<=n;i++){
13            if(i%2!=0){
14                System.out.print(a+" ");
15
16            }
17            sum=a+b;
18            a=b;
19            b=sum;
20
21        }
22
23    }
24 }
```

5



i = 0 ; i < n

HW_Running sum and average

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Emma was given a programming task to create a program that takes an integer input **n** and prints the **running sum** from **1 to n** along with its **average**. Emma created a loop to calculate the **sum and average** and printed the results to the console. The program worked perfectly, and Emma was praised for her coding skills.

Input Format

For each test case, you will get **n** as an integer input.

Constraints

```
1 <= n <= 1000
```

Output Format

First line print the **sum**.

Second line print its **average**.

Sample Input 0

```
5
```

Sample Output 0

```
15
3
```

Explanation 0

First line sum from 1 to 5 is 15.

Second line average is 3.

Submitted Code

```
Language: Java 15

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9         int runSum=0;
10        int avg=0;
11        for(int i =n;i>=0;i--)runSum+=i;
12
13        avg=runSum/n;
14
15
16        System.out.println(runSum);
17        System.out.println(avg);
18
19    }
20
21 }
```

$n = 5$

Running sum $= 5 + 4 + 3 + 2 + 1 + 0 \Rightarrow 15$

Average $= \frac{\text{sum of the total n}}{\text{no of val}}$

$\frac{15}{5} \Rightarrow 3$

