MINET 2007

Programming (Finals)

1. *This paper contains 5 printed pages with 4 questions.*
2. *You have* ***2 hours*** *to attempt the questions.*
3. *The marks for each question are indicated near the bottom of the question.*
4. *When you think you have completed a question, shout "Attempt!" and a judge will come to check your solution.*
5. *If your solution is correct, you will be awarded the marks for the question plus* ***bonus marks*** *which will be numerically equal to the* ***time remaining in minutes*** *when you attempted your question.*
6. *For an* ***incorrect attempt****,* ***¼ of the marks*** *of the question will be deducted.*
7. *If you have any queries, feel free to ask the judges at any time.*
8. *Decisions of the judges will be final and binding on all participants.*

Best of Luck!

Question 1
Library Management

The city library recently got a new stock of books. The cataloguing of the books was done automatically by the library computer, and the labels assigned were the cardinal numbers 1, 2, 3, 4 and so on, without any leading zeroes. The librarian wants to keep a record of the number of books, but due to a bug in the library software, the catalogue does not contain a book count. She needs to submit a report to her manager within a few hours, and does not have the time to count the books manually. She then comes up with an innovative solution.

The printers that provided her with labels to stick on the books charged her per digit printed, and in the invoice they have mentioned the **number of digits**. Thus, she needs to use this number to work out the number of books in the library.

Given the number of digits N printed for the labels ($1 \le N \le 2 \times 10^9$), calculate the number of books in the library. Return -1 if the given number of digits does not correspond to a valid solution.

For example,

- If the number of digits is 11, then the number of books would be 10, because each of the first 9 numbers (1-9) has one digit, and the 10^{th} book has a two-digit label.
- An input of 10 digits should return -1 (no possible solution), because it takes 9 digits to label 9 books and 11 digits to label 10.

Input:
The number of digits N ($1 \le N \le 2 \times 10^9$)

Output:
The number of books in the library

Examples:
Input: 11
Output: 10

Input: 10
Output: -1

Input: 1863927342
Output: 219448716

Input: 1863927343
Output: -1

[Marks: 60]

Question 2
Safe Cracking

A door of a safe is locked by a password. Josh witnessed an employee opening the safe. Here is the information Josh got from spying on the employee.

- The password is a sequence containing exactly *N* digits ($2 \le N \le 30$).
- The password is entered using the keypad shown in the picture below.
- Every pair of neighboring digits in the password **is adjacent on the keypad**. Two digits are adjacent on the keypad if they are distinct and have a common edge. The digits adjacent to 4 are shown in the figure in bold

1	2	3
4	5	6
7	8	9
0		

Josh has evil intentions of unsealing the safe, but before he can realize his plan, he wants to know how many different passwords exist. Given the number of digits *N*, return the number of possible passwords that satisfy the adjacent-key constraint specified above.

For example, if the password contains 2 digits, then the possible passwords, using this keypad, are as follows.

- if the first button is 1, the second button can be either 2 or 4
- if the first button is 2, the second button can be either 1, 3 or 5
- if the first button is 3, the second button can be either 2 or 6
- if the first button is 4, the second button can be either 1, 5 or 7
- if the first button is 5, the second button can be either 2, 4, 6 or 8
- if the first button is 6, the second button can be either 3, 5 or 9
- if the first button is 7, the second button can be either 4, 8 or 0
- if the first button is 8, the second button can be either 5, 7 or 9
- if the first button is 9, the second button can be either 6 or 8
- if the first button is 0, the second button can be 7 only

Thus the total number of 2-digit passwords possible is 26.

Input:
The number of digits *N* ($2 \le N \le 30$) in the password

Output:
The number of possible passwords satisfying the adjacent-key constraint above

Examples:
Input: 2
Output: 26

Input: 25
Output: 768478331222

[Marks: 100]

Question 3
<u>Inside the Ellipse</u>

An ellipse is a figure on a plane where the sum of the distances from any point on its perimeter to two fixed points is constant. The two fixed points are called foci (plural of focus). In this problem we are interested in finding the number of points **with integral coordinates** that lie strictly inside the given ellipse defined by coordinates of foci *(h, k)* and *(x, y)* and the constant sum of distances *d*, which are given as input (-100 ≤ *h, k, x, y* ≤ 100 and 1 ≤ *d* ≤ 200). The ellipse defines a positive area.

For example, the ellipse defined by (0, 0) and (0, 0) with sum of distances 4 is a circle with radius 2. The points (-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0) and (-1, 1) lie strictly inside the circle. The points (-2, 0), (0, -2), (0, 2) and (2, 0) are on the perimeter, so they do not lie strictly inside the circle and should not be counted. The total number of points is 9.

Input:
The first four numbers *h*, *k*, *x*, and *y* correspond to the coordinates of the foci *(h, k)* and *(x, y)* (-100 ≤ *h, k, x, y* ≤ 100). The next number on the same line is the sum of distances from the foci *d* (1 ≤ *d* ≤ 200)

Output:
The number of points with integral coordinates that are strictly inside the ellipse

Examples:
<u>Input</u>: 0 0 0 0 4
<u>Output</u>: 9

<u>Input</u>: -3 0 3 0 10
<u>Output</u>: 59

[Marks: 100]

Question 4
<u>Base Conversion</u>

It is common in modern written languages to express numbers in base 10, where the digits 0-9 are used to express values as sums of powers of 10. For instance, the number written as 12345 in base 10 refers to the sum $1\times10^4 + 2\times10^3 + 3\times10^2 + 4\times10^1 + 5\times10^0$. It is also common in various applications to represent numbers in other number bases (such as base 2, which is how computers represent numbers). Additionally, a minus sign preceding a number denotes that it is negative.

Consider for a minute what it would mean to express a number in a **negative base**, -10. The number written as 12345 in base -10 refers to the sum $1\times(-10)^4 + 2\times(-10)^3 + 3\times(-10)^2 + 4\times(-10)^1 + 5\times(-10)^0$. Note that the odd powers of ten are negative, and the value of that sum is 10000 - 2000 + 300 - 40 + 5, which comes out to 8265 in base +10. It turns out that there is a unique way of representing any integer in negative bases (as long as the absolute value of the base is at least 2). The really interesting thing about using negative number bases is that the minus sign isn't needed to represent negative numbers, for example, -1 can be expressed as 19 in base -10, -2 is 18, and so on.

For this problem, write a program that converts a number x ($-1000000000 \le x \le 1000000000$) to base b ($2 \le b \le 10$), and returns this representation as a string. The minus sign should be used to denote negative numbers in positive bases, but shouldn't ever be present in numbers in negative bases. The output string should have no unnecessary leading zeros.

Input:
The number x ($-1000000000 \le x \le 1000000000$)
The number b ($2 \le b \le 10$)

Output:
The representation of x in the new base b

Examples:
<u>Input</u>: 12345 10
<u>Output</u>: 12345

<u>Input</u>: 8265 -10
<u>Output</u>: 12345

<u>Input</u>: 1001 -2
<u>Output</u>: 10000111001

<u>Input</u>: -52 -2
<u>Output</u>: 11011100

[Marks: 100]