DFS LAB – ASSIGNMENT 1

MTech(CS) I year 2018–2019

Deadline: 10 September, 2018

Total: 60 marks

SUBMISSION INSTRUCTIONS

1. Naming convention for your programs: `cs18xx-assign1-progy.c`
2. To submit a file (say `cs18xx-assign1-progy.c`), go to the directory containing the file and run the following command from your account on the server (IP address: 192.168.64.35)

   ```
   cp -p cs18xx-assign1-progy.c ~dfslab/2018/assign1/cs18xx/
   ```

   If you want to submit your files from a computer with a different IP address, run

   ```
   scp -p cs18xx-assign1-progy.c \
   mtc18xx@www.isical.ac.in:/user1/perm/pdslab/2018/assign1/cs18xx/
   ```

   (enter your password when prompted).

   To submit all `.c` and `.h` files at one go, use `cp -p *.c *.h ~dfslab/2018/assign1/cs18xx/` or similar.

NOTE: All programs should take the required inputs from stdin, and print the desired outputs to stdout.

Q1. We know that the first two Fibonacci numbers are 0 and 1, and the successive Fibonacci numbers are generated by adding the preceding two numbers in the sequence. Now consider another sequence where we are given the first k numbers, and every number after that is generated by multiplying the previous k numbers in the sequence. If $k = 3$, and we are given 1, 2, 3 as the first three numbers, then the 4th, 5th and 6th numbers in the sequence are 6, 36 and 648, respectively. Given n, k, and the first k numbers, your objective is to print the nth number from that sequence. You should try to minimize the number of multiplications needed. [10 marks]

Input Format
The input (to be read from stdin) is a pair of positive integers n and k, and the first k numbers in the sequence (in order).

Output Format
The output (to be printed to stdout) will be the nth number in the sequence.

Sample Input 0
6 3 1 2 3

Sample Output 0
648

Sample Input 1
7 3 1 1 2

Sample Output 1
128

Sample Input 2
6 4 1 2 3 4

Sample Output 2
576

Q2. The *mimic* function in number theory is defined as follows.

Definition For a given positive integer $m > 1$ (called the *base*), a positive integer N, and a divisor d of N, the mimic function $f(N, d)$ with respect to the base m, is given by

$$f(N,d) = \left| \sum_{i=0}^{n} a_i (m-d)^i \right|$$

where a_i is the ith digit in the base-m representation of N.

Note that, the least significant digit is regarded as the 0th digit, and the most significant digit as the nth digit. Thus, $N = \sum_{i=0}^{n} a_i \times m^i$. Let us further assume that $f_m(N, d)$ denotes the base-m representation of $f(N, d)$.

By using this definition of mimic function, the mimic number of any non-prime integer is defined as follows.

Definition For a given base m, a positive integer $N = \sum_{i=0}^{n} a_i m^i$, and a divisor d of N, the mimic number of N with respect to d is defined as the smallest number k for which $f_m^k(N, d) = d$.

Note that $f_m^k(N, d)$ denotes

$$\overbrace{f_m(f_m(\ldots f_m(f_m}^{k \text{ times}}(N,d),d)\ldots),d) \ .$$

Given an integer and its base (no more than 10), find out its mimic number with respect to another integer by writing a program. [15 marks]

Input Format
The input is a set of three integers. The first integer (m) is a base, the second number should be interpreted as a number N written in base m and is the number whose mimic number is to be found out, and the third number is the divisor d (written in normal, base 10 notation). The base m is restricted to no more than 10.

Output Format
The output should be -1 if d is not a factor of N. Otherwise, the output should list $N, f_m(N, d), f_m(f_m(N, d), d), \ldots, d$, followed by the mimic number in the next line.

Sample Input 0
10 855 9

Sample Output 0
855 18 9
2

Sample Input 1
10 113 9

Sample Output 1
-1

Sample Input 2
8 1725 9

Sample Output 2
1725 9
1

Q3. Consider a simple encryption scheme (Vigenère cipher) in which, given a plain text (PT) message, you pick a suitable word from the text and use that as an encryption key to generate the cipher text (CT). Applying the Vigenère cipher can be described algebraically as follows. If the letters A–Z are taken to be the numbers 0–25 ($A \equiv 0$, $B \equiv 1$, etc.), and addition is performed modulo 26, Vigenère encryption E using the key K can be written as

$$C_i = E_K(M_i) = (M_i + K_i) \bmod 26$$

where C_i is the ith character of CT, M_i is the ith character of PT, and K_i is the i-th character of the key.

Example:

```
PT:  T h e  q u i c k  b r o w n  f o x  j u m p s  o v e r  l a z y  d o g s.
Key: b r o  w n b r o  w n b r o  w n b  r o w n b  r o w n  b r o w  n b r o.
CT:  u y s  m h j t y  x e p n b  b b y  a i i c t  f j a e  m r n u  q p x g.
```

As the example shows, the key is repeated as needed and added to the plain text. Notice that only the alphabets are changed, numbers and special characters are kept as they are. Given two files one containing the plain text and the other containing the cypher text your objective is to find out which word of the plain text was used for the encryption. [25 marks]

Input Format
Names of two files containing text (plain text and cipher text, respectively).

Output Format
The key as a single word.

Sample Input 0
Plaintext file contains: `The quick brown fox jumps over 13 lazy dogs`
Ciphertext file contains: `ehd ofibi mrnuy fnv uulnd oucc 13 lzxj dned`

Sample Output 0
lazy

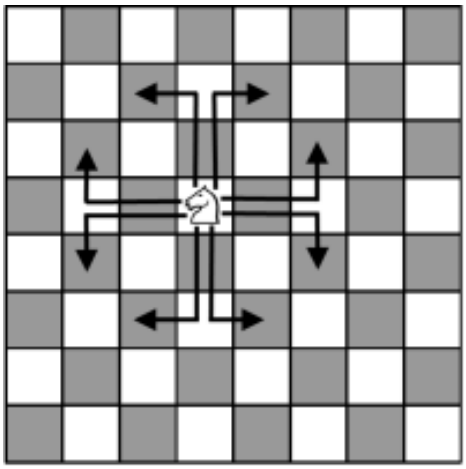
Sample Input 1
Plaintext file contains: `Moorgate got nine men in to get a groom`
Ciphertext file contains: `sfcfsgks uaz ewbq svb wz zf usf g xfcas`

Sample Output 1
groom

Sample Input 2
Plaintext file contains: `The strongest muscle in the body is the tongue`
Ciphertext file contains: `mvr ynvhbtkmx fififi bb gny fhrl om xas guhkns`

Sample Output 2
tongue

Q4. In chess, a knight can move to a cell that is horizontally 2 cells and vertically 1 cell away, or vertically 2 cells and horizontally 1 cell away (see Fig. Q4.). Thus, eight moves are possible for a knight in general.



Given an $m \times n$ chess-like board, and a knight at position (i, j) (where $0 \leq i < m$ and $0 \leq j < n$), compute the probability that the knight remains on the board after a given number (say k) of steps, assuming that, at each step, all eight moves are equally likely. Note that, once the knight moves off the board, it can never return to the board. [10 marks]

Input Format
m n i j k

Output Format
The probability value rounded up to 6 decimal places.

Sample Input 0
1 2 0 1 1

Sample Output 0
0.000000

Sample Input 1
8 8 3 3 1

Sample Output 1
1.000000

Sample Input 2
6 6 0 0 1

Sample Output 2
0.250000

Sample Input 3
4 4 0 0 2

Sample Output 3
0.125000

Explanation for the above

	0	1	2	3
0	0/2/2		2	
1			1	2
2	2	1		
3		2		2/2

The knight starts at the top-left corner (0,0). After 1 move, it may either move off the board (permanently), or go to the squares marked 1 (in red / blue). After another move from the square marked with a red (respectively, blue) 1, it lands on one of the squares marked with a red (respectively, blue) 2, or moves off the board. The knight stays on the board for 8 distinct sequences of moves (of 2 steps each). The total number of such moves of 2 steps each is $8 \times 8 = 64$.