

11주차 실습

2017-11-16

Topic

- Linked List
- Stack

테스트용 서버

- <http://pp2.snucse.org:8888/>
- 6주차부터 현재까지의 Task들이 올려져 있습니다.
- 딜레이 제출시 위 사이트를 이용해 테스트를 하되, 제출은 메일로 하시기 바랍니다.

딜레이 제출 기한

- 원칙적으로는 딜레이를 받지 않으므로, 점수를 받지 못할 수 있습니다.
- 그 다음주 실습 시간 전 (목요일 14:30)까지만 받습니다.
- 그 주 솔루션은 딜레이 제출 기한이 끝난 뒤에 올라갑니다.
 - <https://github.com/ryanking13/2017f-PP-solution>

부분 점수 도입

- 이제 모든 테스트케이스를 통과하지 않고도, 부분 점수를 받을 수 있습니다.
- 하지만, 앞으로 테스트케이스에 대한 직접적인 설명이나 언급은 피할 예정입니다.
 - 보이지않는 문제점을 알아내는 것도 프로그래밍의 **핵심 부분입니다!**

Task 11_1 Crazy Number

Limitation

Time Limit : 1 second / Memory Limit : 16 MB

Description

Every natural number has its ***Crazy Number***, which can be obtained from sum of original decimal number and the number of '1' represented in binary form.

For example, ***Crazy Number*** of 7 is 10, because 7 is converted to 111(2), which has three '1', so $7 + 3 = 10$. In the same way, ***Crazy Number*** of 8 is 9.

Decimal	Binary	# of '1' (binary)	Sum(Crazy Number)
7	111	3	10
8	1000	1	9

Given N Crazy Numbers, find each original number. If you can find 2 or more original number, print the smallest original number.

Task 11_1 Crazy Number

Input

The 1st line of input contains the number of query N ($1 \leq N \leq 100$).

On next N line, each line contains ***Crazy Number*** C ($1 \leq C \leq 10,000,000$).

Output

Output corresponding original number. If original number doesn't exist, print -1.

Example

Input	Output
2	7
10	8
9	
3	-1
4	10
12	-1
6	

Task 11_2 Postfix calculator

Description

Postfix also known as *Reverse Polish Notation* (or *RPN*), is a notational system where the operation/function follows the arguments. [<http://wiki.c2.com/?PostfixNotation>]

For example, we can write $1 + 2 * 3$ to $1\ 2\ 3\ *\ +$ in a postfix notation manner.

It is very useful to make a calculation in machine, because you don't have to consider the operator precedence.

In this problem, you'll going to get an expression written in postfix and print the result of it.

Task 11_2 Postfix Calculator

Inputs

Numbers and operators separated by space. Total number of 'numbers' is less than or equal to **1000**.

Numbers : **digits(0,1,2,...9) only**

Operators : **+, -, *, only.**

Output

The result of postfix expression. Result would be in **integer range**.

Example

Input	Output
1 2 +	3
1 2 3 4 + * -	-13
7 3 * 2 *	42

Task 11_3 Merging Two Array

Description

In this problem, you need to merge two integer arrays.

Two integer arrays, which are already sorted in ascending order, is given.

You need to merge two arrays into one sorted (in ascending order) array.

Inputs

On the first line, N and M, the length of two array, is given. ($1 \leq N, M \leq 100000$)

On the second line, N integers are given. (elements of the first array)

On the third line, M integers are given. (elements of the second array)

Output

Print the merged array in one line, each element separated by space.

Task 11_3 Merging Two Array

Example

Input	Output
3 5 1 10 100 2 8 30 80 120	1 2 8 10 30 80 100 120
5 10 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Task 11_4 Linked List

Description

You will create a program that executes commands related to "Linked List". In this assignment, **no skeleton code is given.** In other words, you have to implement all the parts that you are parsing

Task 11_4 Linked List

MAKE (LIST NAME)

Create a empty Linked List with (LIST NAME).

PUSH_FRONT (LIST NAME) (ELEMENT)

Add (ELEMENT) to front of the linked list whose name is (LIST_NAME).

PUSH_BACK (LIST NAME) (ELEMENT)

Add (ELEMENT) to back of the linked list whose name is (LIST_NAME).

PRINT_FRONT (LIST NAME)

Print the front element of the linked list whose name is (LIST_NAME).

PRINT_BACK (LIST_NAME)

Print the back element of the linked list whose name is (LIST_NAME).

INSERT (LIST NAME) (INDEX) (ELEMENT)

Insert the ELEMENT at the INDEX of the linked list whose name is (LIST_NAME).

DELETE (LIST NAME) (INDEX)

Delete the ELEMENT at the INDEX of the linked list whose name is (LIST_NAME).

Task 11_4 Linked List

CONCAT (LIST A) (LIST B)

Link LIST B after LIST A.

PRINT_ELEMENT (LIST A) (INDEX)

Print the (index)th element of the linked list whose name is (LIST_NAME).

PRINT (LIST A)

Print the elements of A in order.

PRINT_SIZE (LIST A)

Print the number of elements of A.

POP_FRONT (LIST A)

Print the FRONT ELEMENT of A and remove FRONT ELEMENT.

POP_BACK (LIST A)

Print the BACK ELEMENT of A and remove BACK ELEMENT.

Task 11_4 Linked List

Constraints

The value of input is integer.

The number of new LISTS to be created is less than 1000. (LIST_NAME) does not exceed 10 in length.

There is no limit to the size of the linked list and the number of commands. (Only memory size matters.)

Ignore wrong instruction(calling index which is bigger than list size, pop instruction on empty list, or etc.). Just skip that instruction.

Task 11_4 Linked List

Example

input	output
MAKE A	4 3 5
PUSH_FRONT A 3	4 2 3 5
PUSH_FRONT A 4	4
PUSH_BACK A 5	5
PRINT A	2
INSERT A 1 2	
PRINT A	
PRINT_FRONT A	
PRINT_BACK A	
DELETE A 0	
PRINT_FRONT A	

Task 11_4 Linked List

Example

input	output
MAKE A	8
MAKE B	6 5 4 3 16 15 14 13
PUSH_FRONT A 3	6
PUSH_FRONT A 4	13
PUSH_FRONT A 5	4
PUSH_FRONT A 6	
PUSH_FRONT B 13	
PUSH_FRONT B 14	
PUSH_FRONT B 15	
PUSH_FRONT B 16	
CONCAT A B	
PRINT_SIZE A	
PRINT A	
POP_FRONT A	
POP_BACK A	
PRINT_ELEMENT A 1	

메모리, 속도제한

- 메모리
 - `int[1,000,000]` : 4 byte * 1000000 \approx 3.8 MB
- 속도
 - For문이 10억 번 이상 돌아간다면, 무조건 1초 초과.
 - ex) time complexity $O(n^2)$: $n < 100000$
 - ex) time complexity $O(n^3)$: $n < 1000$

테스트케이스 통과가 안될 때

- ✓Parameter, array 등을 올바르게 초기화 했는지
- ✓끝 값(Boundary Condition) 처리가 잘 되어있는지
- ✓값이 큰 input이 들어올 경우의 처리가 잘 되어있는지
- ✓시간 초과의 경우 더 효율적인 알고리즘을 쓸 수는 없는지