Problem A. Team play

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Altynbek, Adilet and Temirlan are engage in sport programming (ACM ICPC). Their skills have rised up since last year, so now they can solve any problem (individually) in A, B and C minutes respectively. To win ACM ICPC semi-final they need to solve N problems. Try to find how many minutes they will spend to solve at least N problems in total.

Input

First line of the input contains one integer N (0 <= N <= 1e12) - number of problems which students must solve. Second line contains three integers A, B, C (1 <= A, B, C <= 1e6) - time that Altynbek, Adilet and Temirlan spend to solve 1 problem individually.

Output

Print the minimum amount of time during which at least N problems will be solved by that team.

Examples

standard input	standard output
100 10 10 10	340
100 1 2 3	55

Note

In the first sample each student solve 34 problem during 340 minutes. 34 + 34 + 34 = 102 which is greater than 100. During 339 minutes each student solve 33 problem which will give 99 in total.

Note, that students can not cooperate to solve any problem.

Hint: Try binary search on the time students will spend (if they can solve in X minutes then they can solve in X + 1 minutes too).

Problem B. Goldbach's conjecture

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Goldbach's conjecture states that each even number starting from 4 can be represented as the sum of two primes. Your task is to verify this conjecture. You are given even integer N. Find primes A and B such that A + B = N.

Input

You are given the only integer N (4 <= N <= 100000, N is even).

Output

Print two primes A and B such that A + B = N.

Examples

standard input	standard output
12	5 7
56	3 53

Note

If there are several possible answers, print any of them.

Hint: Try to obtain all primes up to N using sieve of Eratosthenes. Then try sum of all pairs and print the needed one.

Problem C. Another one easy stack problem

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

In this problem you need to implement complicated version of stack data structure. In addition to standard operations, your stack must store maximum over all values in it. You are given several queries, you must answer them. There are 4 type of queries. "add X"means pushing integer X to stack. "delete"means delete the top element from stack (or nothing if stack is empty). "getcur"means print the top element of stack (or print "error"if stack is empty). "getmax"means print the maximum element of stack (or print "error"if stack is empty).

Input

First line of input contains an integer Q - number of queries (1 <= Q <= 200000). Each of the next Q lines contain one query (1 <= X <= 1000000).

Output

You must answer in each query of type "getcur"and "getmax".

Examples

standard input	standard output
5 add 29704 add 1837 delete getmax getcur	29704 29704
9 add 4318 getmax getcur add 17976 add 20495 getmax delete getcur getmax	4318 4318 20495 17976 17976
2 getcur getmax	error error

Note

Hint: you can use two stacks, first for values and second for maximums

Problem D.

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

В этой задаче вам нужно реализовать почти все стандартные операции на деке и ещё одну дополнительную — *reverse*.

Удачи ;)

Input

Первая строка входного файла содержит единственное, положительное число $Q(1 \leq Q \leq 10^6)$— количество операций.

Последующие Q строк содержат сами операции, по одной стоке в каждой.

- *back*. Вывести последний элемент, удалив его из дека.
- $front$. Вывести первый элемент, удалив его из дека.
- $push\_back\ N$. Добавить число N в конец дека.
- $push\_front\ N$. Добавить число N в начало дека.
- *reverse*. Перевернуть все элементы в деке.

Все числа N целые и по абсолютному значению не преосходят 10^9.

Output

На каждую операцию *back* и $front$ выведите соответствующее число.

Если, на момент выполнения операции, числа в деке не оказалось то выведите ; (без кавычек.

Examples

standard input	standard output
8 *push_back* 20 *push_front* 30 *front* *push_front* 40 *reverse* *front* *back* *front*	30 20 40 ;(
9 *push_back* 1 *push_back* 2 *push_back* 3 *push_back* 4 *reverse* *back* *back* *back* *back*	1 2 3 4