

SUBMISSION FORMAT A1

SUBMIT A FOLDER <ROLLNO>.zip containing

1. <rollnumber>_Q1.cpp
2. <rollnumber>_Q2.cpp
3. <rollnumber>_Q3.cpp

Submission format for question 1:

Submit a single C++ file named <rollnumber>_Q1.cpp with all your code for first question.

Input format for this question:

Just read a single string from stdin.

Output format for this question:

Output a single double value representing the answer after evaluating the input string.

Submission format for question 2:

Submit a single C++ file named <rollnumber>_Q3.cpp.

The following represent the corresponding query number for each function:

1. exponentiation
2. gcd
3. factorial

Your task is to take Q queries as input and perform the corresponding function and display the output.

Input format for this question:

First Line contains the query count Q

Next Q lines contain the query no and input a and b, for factorial single input

Output format for this question:

Q-1 lines displaying queue after each query

NOTE: for correct execution, the first query will be always deque() or deque(n,x)

Output for first query is not required as deque() will be empty

SAMPLE:

INPUT

3

1 2 4

2 3 6

3 6

Output

16

3

720

Submission format for question 3:

Submit a single C++ file named <rollnumber>_Q3.cpp.

The following represent the corresponding query number for each function:

1. push_front(x)
2. pop_front()
3. push_back(x)
4. pop_back()
5. deque()
6. deque(n,x)
7. front()
8. back()
9. empty()
10. size()
11. resize(x, d)
12. clear()
13. D[n]
14. display() - display contents of deque

Your task is to take Q queries as input and perform the corresponding function and display the queue.

Input format for this question:

First Line contains the query count Q

Next Q lines contain the query no and input(if any)

Output format for this question:

Q-1 lines displaying queue after each query

NOTE: for correct execution, the first query will be always deque() or deque(n,x)

Output for first query is not required as deque() will be empty

SAMPLE:

INPUT

6
5
1 2
1 3
3 4
7
10

Output

2
3 2
3 2 4
3
3

