Due: Nov 3 (11:59pm).

Programming Assignment #1

CMSC 351 – Fall 2014

Rules

1) You may only use C/C++, Java.
2) Your program should use the standard input/output. For example C/C++ users should use scanf/printf/cin/cout and Java users should use system.in/system.out. For more information, you may read this: http://en.wikipedia.org/wiki/Standard_streams
3) Your program should not use more than 128MB of memory.
4) For each test case, your program may only run for 5 seconds. Therefore, you need to design efficient algorithms. The runtime requirements are explained in the problem statements.
5) You can only use the basic libraries for reading and storing data. You cannot use any of the sorted containers, linked lists, or algorithms in the standard libraries. You should implement them yourselves. For example, you can use Arrays, ArrayLists, Vectors. However, you may "not" use LinkedList, QList, Sets, Maps, Hashmaps, or sorting algorithms in the libraries.
6) All programs should be coded only by yourself. You may discuss problems with other students as long as there are no written notes. **All the source codes** submitted by the students will be checked with each other **automatically** using a software plagiarism detector. Please be aware of consequences of cheating, fabrication, facilitation, and plagiarism by reading the following website http://www.studentconduct.umd.edu.
7) The programming assignments are going to be graded by an automatic grader. The automatic grader is a program that compares the output of your program for a single test case against the correct output. As the automatic grader is a computer program, it is sensitive to the formatting, spelling, spacing, and etc. of your outputs. In fact, the presentation of your output has to be exactly the same as what is described in the problem statement. The grade for each test case of a problem is only 0 or 1, and there is no partial credit.
8) **Submitting:** For each problem there is a project in the submit server (submit.cs.umd.edu). You will submit only one file for each problem which is the source code. For each problem, the name of the file is specified in the problem statement.
9) If you need more info, please send me an email to mgholami@cs.umd.edu (Milad Gholami) or come to my office hours (Thursdays 2-4).

Practice Problem: Sum

(0% of the total grade)

Submit your program as: sum.cpp or sum.java

Given two numbers a and b, write a program which calculates $a + b$.

Input Description

The first line of the input has two numbers $a < 100{,}000$ and $b < 100{,}000$.

Output Description

The output is a single integer which is $a + b$.

Sample Input

```
12 15
```

Sample Output

```
27
```

Problem 1: Pairs

(1% of the total grade)

Submit your program as: pairs.cpp or pairs.java

Given a sequence of n distinct integers $(a_1, a_2, \ldots, a_n)$ find the number of pairs (a_i, a_j) such that $|2 \times a_i - 3 \times a_j| = k$. For example, if $k = 5$ then list $(1, 3, 2, 7, 5)$ contains 3 such pairs $(2,3), (7, 3), (5, 5)$ such that $|2 \times a_i - 3 \times a_j| = 5$.

Input Description

The first line of the input has two numbers $n < 500{,}000$ and $k < 500{,}000{,}000$. Next line contains n integers $0 \leq a_i \leq 100{,}000{,}000$.

Output Description

The output is a single integer which is the number of pairs (a_i, a_j) such that $|2 \times a_i - 3 \times a_j| = k$.

Grading Guide

If your program running time is $O(n \log n)$, you will pass 100% of the test cases.

If your program running time is worse than $\Theta(nlogn)$, you might pass some test cases based on your running time.

Sample Input

```
5 5
1 3 2 7 5
```

Sample Output

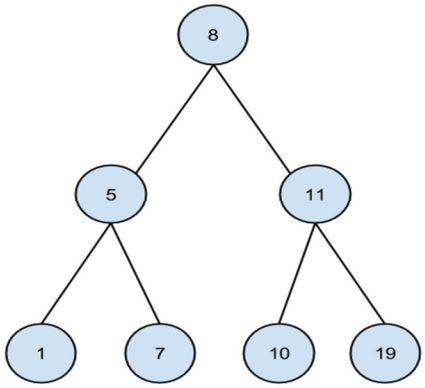
```
3
```

Problem 2: Binary Tree

(1.5% of the total grade)

Submit your program as: btree.cpp or btree.java

Given a preorder traversal of a binary search tree, write a program which calculates the sum of node values at even levels. We define the level of a node as its distance from the root, e.g., root is at level 0. As an example consider preorder traversal (8, 5, 1, 7, 11, 10, 19). This preorder traversal represents the following binary search tree and the sum of node values at even levels is: 8 + 1 + 7 + 10 + 19 = 45.



Input Description

The first line of the input contains $n < 500{,}000$, the number of nodes in the binary search tree. Next line contains *n* numbers which are values of nodes in the preorder traversal of the tree. Each value is an integer less than $500{,}000{,}000$.

Output Description

The output is a single line containing the sum of node values at even levels.

Sample Input

```
7
8 5 1 7 11 10 19
```

Sample Output

```
45
```

Problem 3: Quartile

(Bonus: 2% of the total grade)

Submit your program as: quartile.cpp or quartile.java

A stream of random integers is coming. Write a program which can report the first quartile of the input stream at any point. The first quartile $(Q1)$ is defined as the middle number between the smallest number and the median of the data set when sorted. In particular, the first quartile of a sequence of n numbers would be the $\lceil \frac{n}{4} \rceil$-th smallest element in the sequence.

Input Description

Input consists of multiple lines and each line is a command. There are three types of commands:

1. $i\ x$
2. r
3. end

First command means integer $0 \leq x \leq 1{,}000{,}000{,}000$ is seen in the stream. After *r* your program should report the first quartile of the numbers seen so far. "end" means the end of all commands. The input file has at most $500{,}000$ lines.

Output Description

For each r in the input you should print a number in a line which is the first quartile of all numbers at the time that command r was seen.

Sample Input

```
i 9
i 3
i 6
i 4
i 5
r       //note that the sequence is <9, 3, 6, 4, 5> at this point
i 8
r       //note that the sequence is <9, 3, 6, 4, 5, 8> at this point
i 2
i 1
r       //note that the sequence is <9, 3, 6, 4, 5, 8, 2, 1> at this point
i 10
r       //note that the sequence is <9, 3, 6, 4, 5, 8, 2, 1, 10> at this point
end
```

Sample Output

```
4
4
2
3
```