

Fall 2011 Programming Competition

Run by UPE and RCOS at
Rensselaer Polytechnic Institute

October 30, 2011
12:15 - 2:15

Sponsored By

Bloomberg



0. Introduction

Welcome to this semester's programming competition held by UPE and RCOS. This semester we are lucky enough to be sponsored by Palantir, Bloomberg, and Kitware. To find more details and to submit your solutions please check progcomp.upe.cs.rpi.edu.

Please remember our simple rules:

1. All submitted code must be your own.
2. You are not allowed communicate with/ask questions of any other person.
3. You are allowed to use the internet.

Any attempt to tamper with competition server is considered dishonesty with respect to the contest. Finally, please remember you are always subject to RPI's COMEC policy.

0.1 Questions and Submissions

In this document you will find 4 problems in no particular order. The input and output should be performed using standard in and standard out (with **no ending newline for output**). Here are examples of programs to print the input it receives, line by line in the three supported programming languages C, C++, and Java:

0.1.1 C: test.c

```
#include<stdio.h>

int main(){
    char lines[100];
    scanf("%s",lines);
    printf("%s",lines);
    return 0;
}
```

0.1.2 C++: test.cpp

```
#include<iostream>
#include<string>

using namespace std;

int main(int argc, char** argv){
    string s;
    cin >> s;
    cout << s;
    return 0;
}
```

0.1.3 Java: test.java

```
import java.util.Scanner;

public class submission {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        System.out.print(s);
        return 0;
    }
}
```

Note: The Java class must be named submission.

0.2 Compilation and Execution

All code will be compiled using the following tools. Once compiled, programs will be run by redirecting a file to standard input, and standard output to a file. Programs will have 10 seconds to run. If you have any reason to doubt the grading, notify one of the judges immediately.

0.2.1 C/C++

gcc version 4.2.1 20070719 FreeBSD

0.2.2 Java

```
# java -version java version "1.6.0_07"
Diablo Java(TM) SE Runtime Environment (build 1.6.0_07-b02)
Diablo Java HotSpot(TM) Client VM (build 10.0-b23, mixed mode, sharing)
```

0.3 Important Notes

It is very important that **any code returns 0 from its main**. Any code which does not will get a runerror which is indistinguishable from any other incorrect output error.

1. Goldbach's conjecture

Goldbach's conjecture, http://en.wikipedia.org/wiki/Goldbach%27s_conjecture, states that any even number greater than two can be written as a sum of two primes. Find all such pairs of primes which sum to the even integer given. As in real life, 1 is not considered a prime.

1.1 Input

Input consists of a single even integer N

1.2 Output

The output should on the first line be the number of pairs which sum to N . If $N < 1000$, the following N lines should contain a and b space separated where $a + b = N$, $a < b$ and a and b are prime. The lines should be sorted by the value of a , there should be no repeats. For instance, if the number is 10: 7 3 shouldn't be printed because it repeats 3 7.

If $N \geq 1000$, print the first 1000 a b pairs as defined above.

1.3 Example Input

10

1.4 Example Output

2
3 7
5 5

1.5 Limits

$$4 \leq N \leq 10^6$$

2. Diamond Board Game

You are playing a simple game where you start at the top of a diamond board, and must make your way to the bottom accumulating points, from any position you can only move down right or down left. A board of length 5 looks like the following where a's represent numbers

```
  a
 a a
a a a
 a a
  a
```

Your friend decided that the original game (get the most points) was too easy. His proposed new game is for him to give you a board and an integer x . Your goal is to get as close to the value as possible (minimize $\text{abs}(x - p)$ with respect to your path value p).

2.1 Input

The input consists of the integers N , the board length, and x , the goal number, on the first line space separated. The next N lines contain 1 to $\text{ceil}(N/2)$ space separated integers.

2.2 Output

Your output consists of a single integer p , the value of the path which minimizes $\text{abs}(x - p)$

2.3 Example Input

```
3 -1
-1
2 3
-1
```

2.4 Example Output

```
0
```

2.5 Limits

```
 $N$  is odd
 $1 \leq N \leq 25$ 
 $-10 \leq a \leq 10$ ,  $a$  is an individual value of the board
 $-50 \leq x \leq 50$ 
```

3. Convex Hull

Given a list of points find the convex hull for the points. A convex hull of a list of points is defined as the smallest convex polygon for which all points lie within the polygon. You are to output the area of the convex hull.

3.1 Input

The first line contains N , the number of points. The next N lines contain a single point per line represented as two integers x_i and y_i space separated.

3.2 Output

Your output should contain a single value, the area of the smallest convex hull for the given points.

3.3 Example Input

```
4
0 0
0 2
2 3
1 1
```

3.4 Example Output

```
3
```

3.5 Limits

```
3 ≤ N ≤ 2500
0 ≤ xi, yi ≤ 1000
```

4. Travel Agent

You are interning at a travel agency. They allow customers to book tickets via airplane or bus, and often customers will book transfers between the two travel medium without realizing how agonizing such a trip might be. Your boss is proposing an agony metric for trips which takes into account all of the factors including number of stops, changing medium, time, and cost. An trip's agony is given as the sum of $A_{stop} * NumStops$, $A_{time} * time$, $A_{cost} * cost$, $A_{change} * NumStopsWithChange$.

For instance, if your trip involved taking the bus for an hour and \$10, and then catching a plane for three hours at a cost of \$300. The trip agony would be $A_{stop} + A_{time} * 4 + A_{cost} * 310 + A_{change} * 1$.

The company wants you to implement a way to find the least agonizing trip between two cities given the agony multipliers and a list of potential legs of the trip.

4.1 Input

Your input will contain, on the first line the integers A_{stop} , A_{time} , A_{cost} , A_{change} space separated. The second line will have two city names space separated, the first is the starting city, the second is the ending city. The third line will have an integer N . The next N lines will contain two city names uppercase alphabetic characters (at most 16 characters), and four integers, t_a , t_b , c_a , c_b which represent the travel time for an airplane, travel time for bus, cost of airplane, and cost of bus respectively. These potential paths are directional. All values will always be positive integers.

4.2 Output

Your output should contain a single integer the least agony of any trip. If there is no itinerary print the string 'NONE'.

4.3 Example Input

```
25 10 1 50
CHICAGO DETROIT
1
CHICAGO DETROIT 1 2 150 20
```

4.4 Example Output

```
2 20 40
```

4.5 Limits

```
1 ≤ N ≤ 1000
1 ≤ ta, tb, ca, cb ≤ 1000
0 ≤ Astop, Atime, Acost, Achange ≤ 1000
```