Binghamton University Association for Computing Machinery

Fall 2017 GIM

Who Are We?

- ACM founded in 1947
- World's largest scientific and computing society
- Student chapter
- We love problem solving
- Majors consist of computer science, computer engineering, math, and more

What Do We Do?

- Monthly programming competitions
- International Collegiate Programming Competition (every Fall)
- Host career events and talks
- Hold informative discussions on technical topics
- Technical interview preparation

Programming Competitions

- Platform is hackerrank.com
- You can use basically any language you know
- Open internet
- Problems vary in difficulty
- Food and drinks
- Prizes
- First will be mid to late September

Bloomberg CodeCon

- 5pm - 8pm, Thursday, October 19th in UU 120
- C/C++, Python, Java, C#, Ruby, JS, Go, Rust...
- Food and drinks will be served
- Winner goes to finals in NYC

Sample Problem #1

Alphabetical Ordering

Input Description

You'll be given one lowercase word per line.

```
almost
cereal
```

Output Description

Print the word and whether it is in order or not.

```
almost IN ORDER
cereal NOT IN ORDER
```

Problem #1 Solutions

- Solution 1:
  - Sort word, check if equal to original
  - Issue: It's slow! Sorting is O(n*log(n))
- Solution 2:
  - Run through and check if previous char has a smaller ASCII value than the next.
  - Faster! Only O(n)

Sample Problem #2

Anagram Validation

Input Description

You'll be given two strings, one on each line.

```
debit card
bad credit
```

Output Description

Your program should print ANAGRAM if the strings are anagrams, and NO otherwise.

```
ANAGRAM
```

Problem #2 Solutions

- Solution 1:
  - Sort both strings, check if equal
  - Issue: It's slow! Sorting is O(n*log(n))
- Solution 2:
  - Count frequency of characters to see if both strings have the same characters
  - Faster! Only O(n)

ICPC

International Collegiate Programming Competition

- Greater New York Region
  - Princeton, Cornell, Yale, Columbia, NYU, Stony Brook, etc.
- Teams of 3 (Undergrads and 1st year Grads)
- C, C++, Ada, Java, Python 2 & 3
- We will be trying out candidates and sending up to 3 teams
- One of the teams will be Frosh/Soph

2016 ICPC GNYR Standings

Rank	Name	Solved	Time	A	B	C	D	E	F	G	H	I	J	Total att/solv
1	Princeton 1	10	1006	1/12	1/26	1/32	1/38	1/65	2/113	1/277	2/51	1/205	1/167	12/10
2	Cornell 1	9	693	1/11	1/27	1/16	1/31	1/70	2/127	1/258	1/42	1/111	0/--	10/9
3	NYU 4	8	469	1/11	1/19	1/27	1/22	2/49	2/176	0/--	1/33	1/132	0/--	10/8
4	Yale 1	8	636	1/15	1/42	1/31	1/22	1/97	2/169	1/197	1/63	4/--	0/--	13/8
5	Columbia 2	8	711	1/11	1/49	1/46	1/17	1/60	2/132	0/--	2/33	6/263	0/--	15/8
6	Yale 4	8	981	1/10	1/21	1/51	1/38	1/141	2/235	0/--	1/102	5/283	0/--	13/8
7	Columbia 1	7	646	1/9	1/39	1/63	1/25	1/118	2/299	0/--	1/73	0/--	0/--	8/7
8	Princeton 3	7	665	1/32	1/51	2/65	1/13	1/81	0/--	0/--	2/109	0/--	1/274	9/7
9	Yale 3	7	675	1/9	1/74	1/83	1/37	1/62	2/290	0/--	1/120	3/--	0/--	11/7
10	Princeton 4	7	677	2/38	1/48	1/24	1/47	1/97	2/--	0/--	1/61	4/282	0/--	13/7
11	Cornell 3	7	793	2/26	1/87	1/113	1/27	1/211	1/268	0/--	1/41	0/--	0/--	8/7
12	NYU 3	7	897	1/13	1/35	1/105	1/58	1/224	1/--	0/--	4/185	2/257	0/--	12/7
13	Columbia 4	7	932	1/14	1/33	1/116	1/97	1/198	0/--	0/--	3/232	2/222	0/--	10/7
14	CCNY 1	6	403	2/22	1/45	2/70	1/29	1/95	0/--	0/--	1/122	0/--	0/--	8/6
15	SUNY SB 1	6	438	3/31	1/51	1/137	1/36	1/88	0/--	0/--	2/95	1/--	0/--	10/6
16	Princeton 2	6	678	2/25	1/39	1/65	1/47	1/231	0/--	0/--	2/251	3/--	0/--	11/6
17	NJIT 1	6	686	1/23	1/75	1/128	1/41	1/178	1/--	0/--	1/241	1/--	0/--	8/6
18	Cooper Union 1	5	336	1/11	1/23	0/--	1/97	1/126	0/--	0/--	1/79	0/--	0/--	5/5
19	Yale 2	5	391	1/17	1/51	2/--	1/33	1/134	0/--	0/--	1/156	0/--	0/--	7/5
20	Ramapo 2	5	505	4/140	1/79	1/101	1/18	0/--	0/--	0/--	1/167	1/--	0/--	9/5
21	Binghamton 3	5	508	1/18	1/67	3/198	1/23	0/--	0/--	0/--	2/162	1/--	0/--	9/5
22	Rutgers 1	5	553	1/18	1/158	1/84	1/31	0/--	5/262	0/--	0/--	4/--	0/--	13/5
23	NYU 1	5	585	3/64	3/144	1/153	1/94	0/--	0/--	0/--	1/110	3/--	0/--	12/5
24	Binghamton 1	5	760	3/125	1/144	2/196	1/75	0/--	0/--	0/--	2/200	0/--	0/--	9/5
25	Binghamton 2	4	267	1/12	1/106	0/--	1/88	0/--	0/--	0/--	1/61	0/--	0/--	4/4
26	Cornell 4	4	320	1/45	1/116	3/--	1/86	0/--	0/--	0/--	1/73	1/--	0/--	8/4
27	SUNY SB 2	4	321	1/19	1/60	0/--	1/91	0/--	0/--	0/--	1/151	0/--	0/--	4/4
28	Hofstra 1	4	368	1/26	1/62	0/--	1/93	0/--	0/--	0/--	1/187	0/--	0/--	4/4
29	Cornell 2	4	428	3/117	1/67	1/--	1/45	0/--	0/--	0/--	2/199	0/--	0/--	8/4
30	Columbia 3	4	509	1/33	1/118	0/--	1/174	0/--	0/--	0/--	1/184	0/--	0/--	4/4
31	Cooper Union 3	4	524	3/41	1/64	0/--	1/227	0/--	0/--	0/--	1/192	0/--	0/--	6/4
32	Union College 1	4	566	1/16	2/179	0/--	1/159	0/--	0/--	0/--	1/212	0/--	0/--	5/4
33	NYU 2	4	608	2/39	1/130	0/--	1/188	0/--	0/--	0/--	1/251	0/--	0/--	5/4
34	Marist 2	3	173	1/17	1/51	0/--	1/105	0/--	0/--	0/--	0/--	1/--	0/--	4/3

All but 3 teams solved

A • Which Base is it Anyway?

Programming languages such as C++ and Java can prefix characters to denote the *base* of constant integer values. For example, hexadecimal (base 16) constants are preceded by the string "**0x**". Octal (base 8) values are preceded by the character "**0**" (zero). Decimal (base 10) values do not have a prefix. For example, all the following represent the same integer constant, albeit in different bases.

```
0x1234
011064
 4660
```

The prefix makes it clear to the compiler what base the value is in. Without the "**0x**" prefix, for example, it would be impossible for the compiler to determine if 1234 was hexadecimal. It could be octal or decimal.

For this problem, you will write a program that interprets a string of decimal digits as if it were an octal value, a decimal value or a hexadecimal value.

Input

The first line of input contains a single decimal integer P, ($1 \le P \le 10000$), which is the number of data sets that follow. Each data set should be processed identically and independently.

Each data set consists of a single line of input. It contains the data set number, ***K,*** followed by a single space, followed by a string of at most 7 decimal digits.

Output

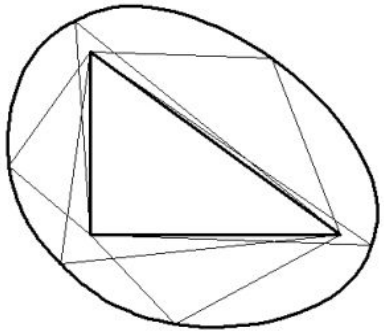
For each data set there is one line of output. The single output line consists of the data set number, ***K***, followed by a space followed by 3 space separated decimal integers which are the value of the input as if it were interpreted to as octal, decimal and hexadecimal respectively. If the input value cannot be interpreted as an octal value, use the value 0.

Sample Input	Sample Output
4 1 1234 2 9 3 1777 4 129	1 668 1234 4660 2 0 9 9 3 1023 1777 6007 4 0 129 297

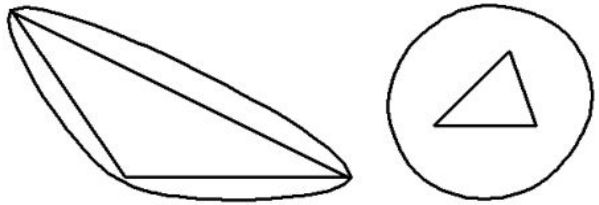
Only two teams solved

J • Smoothed Gardens

The *Flathead Fancy Landscaping Company's* customers are too high-class to have gardens with straight edges so Joe P. Flathead, the owner, has come up with a way to smooth out the contours. He puts a stake in each corner of a triangular plot and drops a loop of rope around the three stakes. Then using a fourth stake in the loop, he pulls the rope tight to mark out a smoothed version of the triangle (see the figure below, the thinner lines are various positions of the rope). This process is similar to the method you learned in middle school to draw an ellipse using 2 push-pins, a piece of string and a pencil, but J.P. Flathead is using *three* stakes (not two), a rope and another stake instead of a pencil.



The longer the rope loop, the smoother the outline will be (see the examples below):



In order to determine how much soil and how many plants are required for the garden, Joe needs to find out the area of the resulting smoothed outline.

For this problem you will write a program which takes as input the coordinates of the corners of the triangle and the length of the rope loop and outputs the area of the smoothed region. The coordinate system will be chosen so that the first vertex is at the origin ($A(0, 0)$), the *x*-axis is along the line from the first vertex to the second vertex ($B(B_x, 0)$) and the final vertex is above the *x*-axis

Weekly Workshops

- Will usually be in workshop style
- Sometimes tech talk style
- ICPC Training
- Mondays from 7:30-9:00pm
- Location TBD

Bloomberg Talk

- 5PM Thursday August 31st
- UU 108
- Learn about Bloomberg and Data Visualization
- tinyurl.com/ybwdonpf

Any Questions?

Sign up for our mailing list here at binghamtonacm.com and like us on Facebook!