University Interscholastic League

**Computer Science Competition**

2014 District Week 1 Programming Problem Set

Judges Packet and Editorials

**Table of Contents**

|  |  |
| --- | --- |
| **Number** | **Name** |
| Problem 1 | Bank |
| Problem 2 | Code |
| Problem 3 | DNA |
| Problem 4 | Exponentiation |
| Problem 5 | Fillings |
| Problem 6 | JSON |
| Problem 7 | M&M's |
| Problem 8 | Rabbits |
| Problem 9 | Right Hand |
| Problem 10 | Stealing Gold |
| Problem 11 | Tale of a Tail |
| Problem 12 | Teams |

**1. Bank**

**Program Name: Bank.java Input File: bank.dat**

==> bank.dat <==

11

Stanley worked for a company in a big building as employee number 427.

Employee 427's job was simple: he sat in room 427 and worked at his desk.

Stanley was happy.

That coffee will cost you $4.20.

Linux L1 3.5.0-45-generic #68-Ubuntu SMP Mon Dec 2 21:58:52 x86\_64 GNU/Linux

2,014 CE

PI, the ratio 4.23 of the circumference of a circle to its diameter, or

3.1415 926 5,35 - and this is just the beginning. It keeps on going,

forever, without 1 2 3 4 ever repeating, which means that contained within

this string 4.3 of decim 09 als is every 03.020 single other number

:D

==> bank.out <==

427

427

427 427

854

-

0

4 20

24

1 3 5 0 45 68 2 21 58 52 86 64

405

2 14

16

4 23

27

3 1415 926 5 35

2384

1 2 3 4

10

4 3 9 3 20

39

-

0

**2. Code**

**Program Name: Code.java Input File: code.dat**

==> code.dat <==

5

zikhmpxjfsvqdaolywtecrbugn 1

2

E UIL Computer Science rocks

D dgr 2014 xplwu

xyikbnmpvaqouctdjszerwlhfg -2

3

E sup bro!

E abcdefg

D 3947

mlnpvjbexoqhzkdgawusrcitfy -18

5

D jeufs

E aft

D efvd

E boat

E cheer

gcnjwsxvlheyztpqfkiumbdaro 23

4

D rhdiq

D amhg

E feelings

D qmhrs nmvt

jgbucsihrmpaqefoyvdlkntzxw -21

10

D tfowj

E animal

E operand

E setting

D nhjqwf

E draper

D lhnwfz

E reaggravation

E anteorbital

D fhotwjn

==> code.out <==

Protocol 1

UIL Cpemdfnx Slgnbln xplwu

uil 2014 rocks

Protocol 2

xpb wqr!

vwgizlk

3947

Protocol 3

grain

ura

rice

tlua

vmdde

Protocol 4

train

word

pttviquf

north pole

Protocol 5

plain

ojwvof

kthaojz

ihqqwjn

gentil

zaotha

begild

ahonnaosoqwkj

ojqhkalwqof

leaping

**3. DNA**

**Program Name: DNA.java Input File: dna.dat**

==> dna.dat <==

7

ATGC

TACG

ATGC

CGTA

AGQ

TCF

CGATAGAT

CCTATCTA

CTTGCTTCGGAAGTCCCGGTGGACC

GAACGAAGCCTTCAGGGCCACCTGG

GGTCGTATCGCT

CCAGCATAGCGG

GTATTAAGGATCCAATTGGTTTCCAATGTATTGAAGCGTCCTCCGGCCATACTCAAGGACGCTGTTAAG

CATAATTCCTAGGTTAACCAAAGGTTACATAACTTCGCAGGAGGCCGGTATGAGTTCCTGCGACAATTC

==> dna.out <==

GOOD

BAD

BAD

BAD

GOOD

BAD

GOOD

**4. Exponentiation**

**Program Name: Exponentiation.java Input File: exponentiation.dat**

==> exponentiation.dat <==

10

5.0 1.0 2

4.5 -7.5 3

-1.0 10 5

2.0 3.0 1

2.0 3.5 2

2.5 4.5 3

0.0 2.0 1

0.1 2.0 3

0.25 2.0 4

3.0 2.0 4

==> exponentiation.out <==

(24.0,10.0)

(-668.25,-33.75)

(-49001.0,90050.0)

(2.0,3.0)

(-8.25,14.0)

(-136.25,-6.75)

(0.0,2.0)

(-1.199,-7.94)

(14.50390625,-7.875)

(-119.0,120.0)

**5. Fillings**

**Program Name: Fillings.java Input File: fillings.dat**

==> fillings.dat <==

10

2 3

2 10

4 9

5 7

3 6

3 9

5 8

3 4

10 10

1 2

==> fillings.out <==

6

90

3024

2520

120

504

6720

24

3628800

2

**6. JSON**

**Program Name: Json.java Input File: json.dat**

==> json.dat <==

17

Cake.ingredient = sugar

print(Cake)

Cake.bake = oven

print(Cake)

dog.treat = bone

print(dog)

boat.type = sailboat

boat.length = 33

boat.color = white

boat.year = 1968

print(boat)

uil.topic = CS

uil.test = 40\_MC

uil.packet = 12

uil.topScore = 1440

uil.status = CS\_Rocks!!

print(uil)

==> json.out <==

{ingredient : sugar}

{ingredient : sugar, bake : oven}

{treat : bone}

{type : sailboat, length : 33, color : white, year : 1968}

{topic : CS, test : 40\_MC, packet : 12, topScore : 1440, status : CS\_Rocks!!}

**7. M&M's**

**Program Name: MandMs.java Input File: mandms.dat**

==> mandms.dat <==

10

1

2

3

5

100

6

7

8

9

101

==> mandms.out <==

1

2

3

4

3

4

3

4

3

4

**8. Rabbits**

**Program Name: Rabbits.java Input File: rabbits.dat**

==> rabbits.dat <==

10

1

2

3

5

14

6

8

30

15

115

==> rabbits.out <==

1

1

2

4

129

6

13

58425

189

7536815746437618530

**9. Right Hand**

**Program Name: RightHand.java Input File: righthand.dat**

==> righthand.dat <==

6

3

...

.##

.#.

5

.###.

...#.

.###.

...##

.#...

10

.....#..##

#.#..###.#

#...#.#...

.#.##.##..

....#..##.

..#....#..

.####.#.#.

#.#.#.##..

.........#

.#.....#..

5

.....

...#.

.###.

#..##

.#...

20

..#....####......#..

#..#..#.#..#.#...#..

#.##...#.##...#...##

....###...#.##.##..#

...#...##...#...##..

......#..#....#.....

..#.##..#.###..#....

#...##......##...#..

...##.....##..#....#

#..##..#.##..#...#..

.#.......#...##.....

.....##.#..##.#.#.#.

#.#.#..#.##...##...#

#.##.#..............

#.#..##....#........

..#...##.#........##

#.#...#.#.#....#....

#..##....##...###..#

.......###..#.....#.

###...#.#.###.#.##..

25

...#...#...##...#..#.#..#

#..#..#.#...##..##..##...

.#..#.........#.......##.

...#..#.....##.#.####....

##......#..##.#....#..##.

...#...###...###.#..#....

##.####..#.#...####.....#

.#....#....#..#..###.....

####.#.....#.#.....#...#.

#.#....#..#...###....#.#.

#..#...####.#.....#....##

....#.###.###.......##.##

#.#.#....#.#...####.....#

#....#..#..#..#.......#.#

..#....#...#.##....#.##..

..#.####.....#..##......#

......##...#....#........

.....#...#..###....#.#...

##...#.#........#.##...##

###......#..#....##.#..#.

.##..............##.#.#.#

.#..#.###.#.###...#.....#

..##...####...#...#.#....

...#......#.######..#.#..

#.#...#...#........##..#.

==> righthand.out <==

NO

YES

YES

NO

NO

YES

**10. Stealing Gold**

**Program Name: Stealing.java Input File: stealing.dat**

==> stealing.dat <==

10

3 2 10

35 68 42

25 70 1

63 59 79

46 6 65

62 28 82

43 96 92

92 37 28

54 3 5

22 83 93

==> stealing.out <==

30

42

1

237

975

410

92

140

175

93

**11. Tale of a Tail**

**Program Name: Tale.java Input File: none**

==> tale.out <==

This is a

very long

and un-

happy

tale

.

**12. Teams**

**Program Name: Teams.java Input File: teams.dat**

==> teams.dat <==

20

11 98

16 149

31 116

12 40

53 14

23 187

14 163

27 21

26 31

19 42

18 56

21 1

20 171

22 85

15 171

17 192

25 49

24 84

29 191

28 34

==> teams.out <==

98

51

131

127

157

30

133

112

455

123

179

178

349

264

435

285

578

494

305

271

University Interscholastic League

**Computer Science Competition**

2014 District Week 1

Problem Editorials

**Table of Contents**

|  |  |
| --- | --- |
| **Number** | **Name** |
| Problem 1 | Bank |
| Problem 2 | Code |
| Problem 3 | DNA |
| Problem 4 | Exponentiation |
| Problem 5 | Fillings |
| Problem 6 | JSON |
| Problem 7 | M&M's |
| Problem 8 | Rabbits |
| Problem 9 | Right Hand |
| Problem 10 | Stealing Gold |
| Problem 11 | Tale of a Tail |
| Problem 12 | Teams |

****

Process the text line by line to identify numbers. You may use the sample solution's methods or Character.isDigit() etc.

**2. Code**

Overview of problem:

We have to manipulate a string character by character according to a given function. We have to look up the character in a map to do the substitution, then math on the ascii value to produce the correct rotation.

Analysis of problem:

Since this translation creates a one-to-one mapping of characters, if we figure out the mapping for encoding, we can create a reverse mapping for decoding without actually having to calculate anything.

Solution:

For each protocol, we first construct the encoding mapping using the substitution string and the offset. Then we can construct a decoding mapping by reversing the encoding mapping. Then, for each phrase we need to encode or decode, we just substitute each character in the string by the one it maps to in the specified mapping.

Constructing the mapping is O(26) for each mapping, and doing the encoding/decoding is O(length of string to encode or decode).

















