J221 - Bus Route Category

Kelvin Chow {Lrt1088}

2022-01-29

香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

Background

Author: Kelvin Chow

Preparation: Susanna Chan, Kelvin Chow, Anson Ho and Kevin Lee

Statistics

Task	Attempts	Max	Mean	Std Dev
J221 - Bus Route Category	81	100	68.444	39.708

Subtasks

7: 72	12: 68	19: 56	21: 55	26: 50	15: 47

SUBTASKS

	Points	Constraints
1	7	The bus route number consists of a number between 1 and 99 (inclusive) onl
2	12	The bus route number consists of a number only
3	19	The bus route number does not consist of a Letter Prefix
4	21	Category A, Category B and Category D must not be `Normal`
5	26	The bus route number is guaranteed to be valid
6	15	No additional constraints

First Solve: dbsjkjk - 0:17

Problem

Given a string in format of [C_1][D_1][D_2]D_3[C_2], where C_i is a Letter and D_j is a digit, items inside a pair of [] are optional.

For example, the input can be 1, 11, 101, A1, 1X, B1P, Z999Z...

According to the tables, find out the Categories C_1, C_2, D_1 and D_2 were represented.

Output the Categories in order of D_2, D_1, C_2, C_1.

Part A

Letter Prefix	Category A
No Letter Prefix	Normal
A	Airport
B	Border
N	Overnight
Other letters	Invalid

4

Part B

Hundreds Digit	Category B
No Hundreds Digit	Normal
1	Cross River
2	Air-conditioned
3	Holiday
Other numbers	Invalid

2

Part C

1

Tens Digit	Category C
0, 1, 2 or No Tens Digit	Downtown
3, 4	West District
7	North District
9	East District
Other numbers	Invalid

Part D

3

Letter Suffix	Category D
A, B, C or No Letter Suffix	Normal
P	Peak Hour
S	Special
X	Express
Other letters	Invalid

Problem

Special Rules:

if (any of the Category is `Invalid`)

output `Invalid` and exit

if ({C_1, C_2} **or** {D_1, C_2} is an invalid pair)

output `Invalid` and exit

Invalid category example: Z987Z
Z - NOT in the table!

Part A	
Letter Prefix	Category A
No Letter Prefix	`Normal`
`A`	`Airport`
`B`	`Border`
`N`	`Overnight`
Other letters	Invalid

4

Part B	
Hundreds Digit	Category B
No Hundreds Digit	`Normal`
`1`	`Cross River`
`2`	`Air-conditioned`
`3`	`Holiday`
Other numbers	Invalid

2

Part C	
Tens Digit	Category C
`0`, `1`, `2` or No Tens Digit	`Downtown`
`3`, `4`	`West District`
`7`	`North District`
`9`	`East District`
Other numbers	Invalid

1

Part D	
Letter Suffix	Category D
`A`, `B`, `C` or No Letter Suffix	`Normal`
`P`	`Peak Hour`
`S`	`Special`
`X`	`Express`
Other letters	Invalid

3

Invalid pair example: 333P
3 - Holiday
P - Peak Hour

Incompatible Pair		
1.	`Overnight`	`Peak Hour`
2.	`Holiday`	`Peak Hour`

Problem

Special Rules:

if (all C_1, D_1 and C_2 are representing `Normal`)

output `Normal` once only

else

Do not output `Normal`

Example: 1
No D_2 - Downtown
No D_1 - Normal
No C_1 - Normal
No C_2 - Normal

Answer: Downtown Normal

Part A	
Letter Prefix	Category A
No Letter Prefix	Normal
A	Airport
B	Border
N	Overnight
Other letters	Invalid

4

Part B	
Hundreds Digit	Category B
No Hundreds Digit	Normal
1	Cross River
2	Air-conditioned
3	Holiday
Other numbers	Invalid

2

Part C	
Tens Digit	Category C
0, 1, 2 or No Tens Digit	Downtown
3, 4	West District
7	North District
9	East District
Other numbers	Invalid

1

Part D	
Letter Suffix	Category D
A, B, C or No Letter Suffix	Normal
P	Peak Hour
S	Special
X	Express
Other letters	Invalid

3

Subtask 1

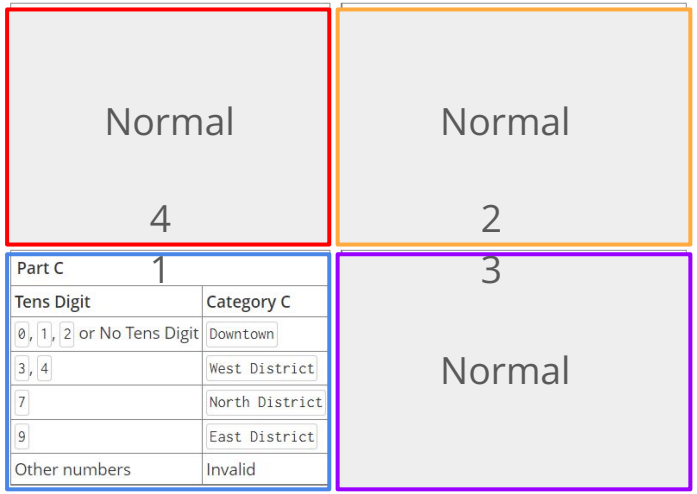
The string is in format of [D_2]D_3. i.e. 1 to 99

You may either read the input as an integer, and divide the integer by 10 to extract D_2, or

Read the input as a string S,

```
if (length(S) == 1)
    D2 = 0
else
    D2 = S[0]
```

1 != '1'!!!

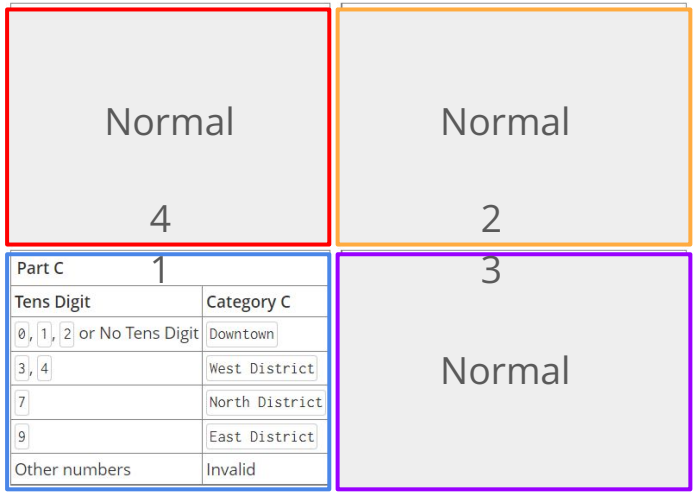


香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

Subtask 1

Because there are no C_1, C_2 and D_1, they all are Normal.

So output "Category-of-D_2 Normal"

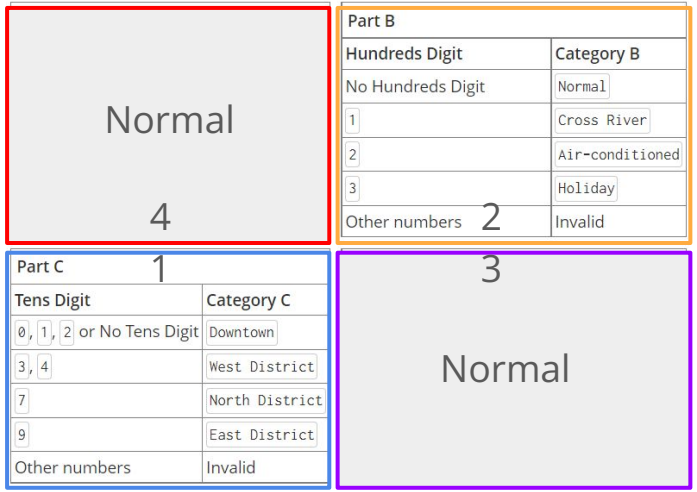


Subtask 2

The string is in format of [D_1][D_2]D_3. i.e. 1 to 999

If you are using String, you need to consider length = 1, 2 or 3, or use integer and then perform /100 to extract D_1, %100/10 to extract D_2.

Output Category-of-D_2 Category-of-D_1



Subtask 3

The string is in format of [D_1][D_2]D_3[C_2]. e.g. 1, 11, 101, 1A, 22Z...

You may read the input as a String S, and then check the last character of S. If the last character of S is a letter, extract it as C_2 and erase it.

Then extract the digits from the string, you may use the same method as Subtask 2.

Remember to check for the invalid pair of {D_1, C_2},

e.g. 301P.

Normal

4

Part B

Hundreds Digit	Category B
No Hundreds Digit	Normal
1	Cross River
2	Air-conditioned
3	Holiday
Other numbers	Invalid

2

Part C 1

Tens Digit	Category C
0, 1, 2 or No Tens Digit	Downtown
3, 4	West District
7	North District
9	East District
Other numbers	Invalid

Part D 3

Letter Suffix	Category D
A, B, C or No Letter Suffix	Normal
P	Peak Hour
S	Special
X	Express
Other letters	Invalid

Subtask 3

Actually, if you are using `<stdio.h>`, you may just use `scanf("%d%c",&a,&b)`.

If there are no C_2, b will be read as '`\n`'. (or '`\r`' in Windows :( )

Subtask 4

Since all categories are not Normal, the input must consist of C_1 (Letter prefix), D_1 (Hundreds Digit) and C_2 (Letter Suffix).

So the string is in format of $C_1D_1D_2D_3C_2$.

You may read the input as a string S and then,

C_1 = S[0], D_1 = S[1], D_2 = S[2], C_2 = S[4]

Remember to check for invalid pairs.

Part A	
Letter Prefix	Category A
~~No Letter Prefix~~	~~Normal~~
A	Airport
B	Border
N	Overnight
Other letters	Invalid

4

Part B	
Hundreds Digit	Category B
~~No Hundreds Digit~~	~~Normal~~
1	Cross River
2	Air-conditioned
3	Holiday
Other numbers	Invalid

2

Part C	
Tens Digit	Category C
0, 1, 2 or No Tens Digit	Downtown
3, 4	West District
7	North District
9	East District
Other numbers	Invalid

1

Part D	
Letter Suffix	Category D
~~A, B, C or No Letter Suffix~~	~~Normal~~
P	Peak Hour
S	Special
X	Express
Other letters	Invalid

3

Subtask 5 and Full solution

Full solution is similar to Subtask 3, you need the consider first character being a letter or not, to extract and erase.

Subtask 5 act as a safety net if you miss some cases.

Subtask 5 and Full solution

(1) Parse the input string in the format of $[C_1][D_1][D_2]D_3[C_2]$
(2) For each of the $C_1D_1D_2C_2$, write if-statements to check its validity and determine the category
(3) Extra checking on incompatible categories
(4) Eliminate Extra “Normal”s when outputting the final answer

How to extract and erase actually?

In C++, string has a method `erase()`, which receive the position and erase it.

https://en.cppreference.com/w/cpp/string/basic_string/erase

And since C++11, there are `std::stoi()`, with receive a string and return the integer value.

https://en.cppreference.com/w/cpp/string/basic_string/stol

Python: `[1:]` and `[:-1]`, `int(str)`

Java: `String.subString()`, `Integer.parseInt()`

How to extract and erase actually?

Or if you do not remember any of the functions, you may scan though the string.

By using two if-statements, extract the letters and indicate the start and end of the digits.

To convert a digits string to an integer:

```
int x = 0
for every digits do
    x = x * 10 + value-of-digits
```

By this algorithm, the digits are converted into a integer x.