MATRIKS

MULTI PRIORITY QUEUE

In this assignment you are expected to develop an extensible system that can sort items according to different type of priorities. An item in the queue can have one or multiple priorities, final priority score of the item can be calculated as a sum of each priority type depending on their own rules. Items can be added, edited or removed; queue must be always sorted after changes.

DESIGN PRINCIPLES/MOTIVATION

Queue and ordering can be easily achieved in current high-level languages. This assignment proposes a problem with an extensible priority management requirement. Therefore your code should be ready for integrating a new Priority Type. Even though the execution of the submission is important, more focus will be on the design of such a simple system. Before submitting your source code, define another priority type to your liking and analyze how many classes are affected and should be changed. If the changes are high, reconsider your design. Even though queue and priority can be easily associated with performance and implementation of such system, this assignment is not focused on a queue implementation from scratch. Linq and standard Collection libraries are free to use.

COMMAND TYPES

All input lines start with a single capital letter as a command parameter and followed by an integer id for following the item. Add and Edit commands continue with several priority properties, which will be described on next section.

A 1 [priority types]	Add new item with Id 1
E 2 [priority types]	Edit item id = 2 with new priority types
R 7	Remove item with id = 7

PRIORITY TYPES

TIME PRIORITY

An item with a time priority can be supplied with T argument followed by an integer. Following integer is the number of minutes. Items with time priority smaller than 5 minutes have priority score of 10, between 5 and 30 minutes priority score is 3, otherwise 1.

Examples;

A 1 T 4	Time priority with four minutes	Score 10
A 2 T 26	Time priority with 26 minutes	Score 3
A 3 T 48	Time priority with 48 minutes	Score 1

ALPHA PRIORITY

Alpha priority can be defined with an A and a char value only consisting of capital English alphabet letters. In alphabetical priority, A has the lowest priority of 1 and Z has the highest priority of 26. All the letters in between have priorities equal to their order in the English alphabet.
Examples;

A 1 A A	Alpha priority with A	Score 1
A 2 A Z	Alpha priority with Z	Score 26
A 3 A J	Alpha priority with J	Score 10

COLOR PRIORITY

Color priority can be defined with a C and a string value representing the color. Three valid color values exist; Red = 20, Green = 4, Blue = 18.
Examples;

A 1 C Blue	Color priority with Blue	Score 18
A 2 C Green	Color priority with Green	Score 4
A 3 C Red	Color priority with Red	Score 20

MULTI PRIORITY

Any item can have multiple priority properties. Priority properties don't have any order. Following examples are all valid.

A 1 C Blue T 4	Item with both Blue (Color) and 4 (Time) priority	Score = 18 (C) + 10 (T) = 28
A 2 A T C Green	Item with both T (Alpha) and Green (Color) priority	Score = 20 (A) + 4 (C) = 24
A 3 A J T 27 C Red	Item with J (Alpha), 27 (Time) and Red (Color) priority	Score = 10 (A) + 3 (T) + 20 (C) = 33

INPUT-OUTPUT

Each command read from the input file should be written to output file on execution. And after each command the current queue should be written to output. When printing the queue each element in the queue should be written on their own line with their Id and current score. i.e. 1 10 line on output means, item with Id 1 has a current score of 10. You can find two sample input files and outputs in the following tables for reference.

INPUT-1

Input-1	Output-1
A 1 T 2 A 2 C Blue A 3 A H	A 1 T 2 1 10 A 2 C Blue

A 4 C Red A 5 T 16 E 1 T 7 R 4 R 3 R 2 A 2 T 7	2 18 1 10 A 3 A H 2 18 1 10 3 8 A 4 C Red 4 20 2 18 1 10 3 8 A 5 T 16 4 20 2 18 1 10 3 8 5 3 E 1 T 7 4 20 2 18 3 8 5 3 1 3 R 4 2 18 3 8 5 3 1 3 R 3 2 18 5 3 1 3 R 2 5 3 1 3 A 2 T 7 5 3 1 3 2 3

INPUT-2

Input-2	Output-2
A 1 C Blue T 4 A 2 A T C Green A 3 A J T 27 C Red E 2 A Z C Green	A 1 C Blue T 4 1 28 A 2 A T C Green 1 28 2 24 A 3 A J T 27 C Red 3 33

	1 28 2 24 E 2 A Z C Green 3 33 2 30 1 28

Input files will be free of errors, do not error check in your source code. Your user is sane. All commands will be inside an input file and the outputs are expected to be written on an output file. Filenames will be provided as a command line argument. Input files are plain txt files. Application will be called from command line as followed;

```
./queue.exe input-1.txt output-1.txt
```

SUBMISSION

You should submit your C# source code with a Visual Studio solution file. Do not submit any executable files, clear your project before submission. Send all project files to careers.commodity@matriksdata.com in a compressed .rar file. If you have any problem on submission, please contact with an email. With your submission, provide at least a paragraph of description on your source code and your design decisions.

DISCLAIMER

This assignment is intended only for the person to whom it is addressed. Sharing the definition or source codes of the assignment with any other person or on a public platform is prohibited. If this assignment is not sent to you from careers.commodity@matriksdata.com email address, please contact the email as soon as possible.