

The Crawford Technologies Programming Challenge C/C++ #2

# Purpose

The purpose of this challenge is to gather further evidence of the true abilities and qualifications of potential candidates for employment in Crawford Technologies software engineering division than would normally be possible via a simple interview.

The prospective employee is asked to take the enclosed project requirements and return to Crawford a functionally complete, working piece of software, for review by Crawford Technologies Management and Technical staff. Crawford’s staff will use this to evaluate the design decisions made by the prospective employee, as well as the craftsmanship and quality of the code and the project returned. This project will have a significant impact on the applicant selection process. Please take this opportunity to demonstrate for Crawford Technologies your skills in software engineering.

# The Challenge

One of the files Crawford Technologies is dealing with are symbolset files. The file contains a mapping from characters id-s of a certain font to their corresponding Unicode values. Character id is a number between 0 and 255 representing 255 possible characters of the font. It appears in hexadecimal representation, i.e. between 0x00 and 0xFF. Unicode value provides the meaning to the character, for example, Unicode of character ‘B’ is 0x42, and of ‘⅔’ symbol is 0x2154. Note that both are represented in hexadecimal. Knowing the Unicode, screen readers can identify and read the text properly.

The file consists of several lines, each line should be either a comment: in this case the line will begin with ‘#’ character, or a valid line mapping character to its Unicode.

For example:

#some comment

0x98 0x02DC #SMALL TILDE

Note that the Unicode is separated from the character id by one or more blank characters and *can* be followed by a comment usually specifying the name of the character.

You will be given a symbolset file and will have to write a function that will read it and then print out the Unicode values of characters 0x30, 0x40 and 0x50.

The output should be like

*Character ID: 0xNN, Unicode: 0xNNNN*

Where 0xNN is the character ID, 0xNNNN is the Unicode value.

If character with that ID doesn’t exist in the symbolset, output

*Character 0xNN doesn’t exist*

Afterwards, it will look for characters having Unicode 0x20, 0x2154, and 0x913, if such a character exists, it will output it, otherwise, it will output an error message.

Some points to note:

1. Fonts may contain less than 255 chracters, so some character ids may be not occupied
2. The function is for testing purposes only, so write code in a way that will allow other modules to parse the Symboset file, to identify the Unicode of different characters and search for font characters having certain Unicode values.

For example, we want to have a function *getUnicodeValue(int characterId),* which can be either a class member or standalone which will return either Unicode value or 0 if not found.

We also would like to have a function *getCharacterId(int UnicodeValue)*.