100

September 6, 2017.

# Instituto Tecnológico y de Estudios Superiores de Monterrey

**Campus Santa Fe**

**Escuela de Ingeniería y Ciencias, Región Ciudad de México**

**Departamento de Computación**

## Compiler Design First Exam

Instructor: Ariel Ortiz

Course Number and Section: Tc3048.1

Name: Jonathan Kunstmann Nagel

Student ID: A01021617

*Apegándome al Código de Ética de los Estudiantes del Tecnológico de Monterrey, me comprometo a que mi actuación en este examen esté regida por la honestidad académica. En congruencia con el compromiso adquirido con dicho código, realizaré este examen de manera honesta y personal, para reflejar, a través de él, mi conocimiento y aceptar, posteriormente, la evaluación obtenida.*

Firma: _____

**VERY IMPORTANT:** This exam should be solved individually. Any type of plagiarism or copying will be sanctioned with a grade of 1/100 and will also be reported to the institution's Academic Integrity Committee. This sanction is for both the person who copies and for the person who allows to be copied.

## General Instructions

Download the exam related file `exam1.zip` from the course web site. You must only modify the files called `problem1.cs` and `problem2.cs`. **Type your name and student ID in a comment at the top of these source files.**

The corresponding executable files will be built using the provided `Makefile` and tested using the `example1.f` and `example2.html` files.

Once you've finished your exam, create a compressed ZIP file containing only the `problem1.cs` and `problem2.cs` source files. Name this file `solution1.zip`. Upload the ZIP file using the course website.

**Time limit:** 90 minutes.

100

## Problems

50 1. **(50%)** Write in the `problem1.cs` source file a C# program that uses the .NET Framework regular expression API. Your program must read a Fortran 77 source file and print it to the standard output but with all its line comments removed. The name of the input file must be provided as a command line argument.

   ANSI X3.9-1978 (The Programming Language Fortran 77 standard) states in section 3.2.1 that a *comment line* is any line that contains a "C", "c", or "*" character in column 1. Comment lines may appear anywhere in the program unit and are ignored by the compiler.

Example:

```
$ mono commentstrip.exe example1.f
      program pi
      integer numrects, i
      real mid, height, width, area, sum
      sum = 0
      write(*, *) 'Number of rectangles:'
      read(*, *) numrects
      width = 1.0 / numrects
      do 42 i = numrects - 1, 0, -1
          mid = (i + 0.5) * width
          height = 4.0 / (1.0 + mid ** 2)
          sum = sum + height
42    continue
      area = width * sum
      write(*, *) 'Computed pi = ', area
      stop
      end
```

50 2. (50%) A *numeric character reference* (NCR) is a common markup construct used in SGML and SGML-derived markup languages such as HTML and XML. It consists of a short sequence of characters that, in turn, represent a single character. Unicode code points are typically used for this purpose. For example, the Greek capital letter Sigma Σ (code point U+03A3) can be represented as "Σ" or "Σ" in HTML5 documents. Note that, in this case, the first NCR uses base 10 (decimal) and the second NCR uses base 16 (hexadecimal). In other words, the "x" after the "#" symbol indicates an NCR with an hexadecimal code.

Write in the `problem2.cs` source file a C# program that uses the .NET Framework regular expression API to search for all the base 16 NCRs contained in an HTML document and replaces them with with their equivalent base 10 NCRs. The name of the input file must be provided as a command line argument.

Example:

```
$ mono hexreplace.exe example2.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Example 2 HMTL</title>
  </head>
  <body>
    <h1>&amp;#171;An&amp;#32;Example&amp;#187;</h1>
    <ol>
       <li>&amp;#128526; SMILING FACE WITH SUNGLASSES</li>
       <li>&amp;#128520; SMILING FACE WITH HORNS</li>
       <li>&amp;#9829; BLACK HE&amp;#65;RT&amp;#32;SUIT</li>
       <li>&amp;#9752; SH&amp;#65;MROCK</li>
    </ol>
  </body>
</html>
```

Your regular expression is: &#x[a-fA-F0-9]*

↑ should be "+"