**HOMEWORK 4– CS 347 Spring 2010**

**Due: May 19, 2010**

(60 pts)

**Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Names of students you worked with on this assignment (if any): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

You are encouraged to work with other students on homework assignments. You must turn in separate programs and indicate with whom you worked; indicate this in the block comment at the beginning of your program.

**You must turn in**:

* **Source code listings** (**a hardcopy**) for each program.
  + Use the Application -> Programming -> KDEVELOP programming environment to print your assembly source code.
* On **Blackboard turn in** your **three .asm files**.

Be sure to include “self-contained” documentation in your source code, i.e., appropriate comments and explanations of the code. The program will be graded for correct execution and quality of the code and documentation.

**See the grading rubric posted on Blackboard.**

1. Write a procedure that receives a string via the stack (i.e., **the string pointer** is passed to the procedure) and removes all leading and duplicate blank characters in the string. Input strings should be a maximum of 80 characters. For example, if the input string is (\* indicates a blank character):

\*\*\*\*\*Now\*\*\*is\*the\*\*\*\*\*\*\*\*\*\*\*\*time

the output string will be:

Now\*is\*the\*time

Split the source code into two modules. Call the source code **mstrebl.asm,** and **strebl.asm**., for the main program and the string removal procedure. (30 pts)

1. Write a program that requests a string and a substring from the user and reports the location of the first occurrence of the substring in the string. Write a C main program to receive the two strings from the user. Have the C main program then call an assembly language procedure to find the location of the substring. This procedure should receive two pointers to strings string and substring and search for substring in string. If a match is found, it returns the starting position of the first match. Matching should be case sensitive. A negative value should be returned if no match is found. For example, if

string = Good things come in small packages.

and

substring = in

the procedure should return 8 indicating a match of in in things.

Name the source code files: **substrng\_a.asm, substrng\_c.c**. (30 pts)

Be careful about input in C. Be sure you have what you think you have before invoking the assembly procedure.