CSE 320 **Fall 2018**

Computer Project #10

Assignment Overview

This assignment develops familiarity with data structures in assembly language. You will develop an ARM assembly language function to complete a program which manages statistics for a hockey team.

It is worth 10 points (1% of course grade) and must be completed no later than 11:59 PM on Tuesday, 11/20.

Assignment Deliverables

The deliverables for this assignment are the following files:

`proj10.makefile` – the makefile which produces "proj10"
`proj10.support.s` – the source code for your support module

Be sure to use the specified file names and to submit them for grading via the CSE handin system.

Assignment Specifications

The program will use an ordered table to maintain the data set, where each player's jersey number will serve as a unique key to identify that player. The capacity of the ordered table will be determined when it is created.

1. The instructor-supplied driver module (function "main" and associated functions) will perform all input and output, and will manage the overall operation of the program.

2. You will supply the function whose declaration is listed below:

```
int search( struct table*, unsigned long, struct player** );
```

That function (and any "helper" functions which you develop) will constitute a module named "proj10.support.s".

Assignment Notes

1. The functions in your support module must be hand-written ARM assembly language functions (you may not submit compiler-generated assembly language functions).

2. The file "project10.support.h" (below) includes all relevant declarations, along with descriptive comments.

3. The file "project10.driver.o" contains the instructor-supplied driver module.

4. The file "project10.data" contains a sample data set (the statistics for the MSU Men's Hockey team during the 2017-2018 season). Your program must function correctly for that sample data set, as well as any other properly formatted data set.

5. You may wish to create a stub for the required function, then translate, link and execute the program to explore the behavior of the driver module.

6. You will design and implement additional functionality in subsequent projects, so you would be wise to properly structure and comment your source code.

```
/*************************************************************************/
/*  Declarations for Project #10                                         */
/*************************************************************************/


struct player
{
  unsigned short number;    /* player's jersey number (key) */
  char name[25];            /* player's name                */
  unsigned short games;     /* number of games played       */
  unsigned short goals;     /* number of goals scored       */
  unsigned short assists;   /* number of assists scored     */
  unsigned short points;    /* points (goals + assists)     */
  float points_per_game;    /* points per game played       */
};


struct table
{
  unsigned short capacity;  /* number of elements in table  */
  unsigned short count;     /* number of players in table   */
  struct player* memory;    /* pointer to array of players  */
};


/*************************************************************************/
/*  Function:  search                                                    */
/*                                                                       */
/*  Purpose:  locate and return a pointer to a player, if the player     */
/*  is present in the table.                                             */
/*                                                                       */
/*  Arguments:                                                           */
/*    pointer to table of players                                        */
/*    jersey number of player to be located                              */
/*    pointer to pointer to player                                       */
/*                                                                       */
/*  Return value:                                                        */
/*    1 (true) if player located, 0 (false) otherwise                    */
/*************************************************************************/

int search( struct table*, unsigned long, struct player** );
```