CSE 320 **Fall 2020**

Computer Project #11

Assignment Overview

This assignment develops familiarity with data structures in assembly language. You will develop a set of ARM assembly language functions to complete a program which manages student statistics.

It is worth 50 points (5% of course grade) and must be completed no later than 11:59 PM on Thursday, 12/10.

Assignment Deliverables

The deliverables for this assignment are the following files:

`proj11.support.s` – the source code for your support module
`proj11.makefile` – the makefile which produces **`proj11`**

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 statistics about a set of students, where each student's identification number will serve as a unique key to identify that student. 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 functions whose declarations are listed below:

```
int search( struct table*, unsigned long, struct student** );
int delete( struct table*, unsigned long );
int insert( struct table*, unsigned long, char*, int, int, int );
```

Those three functions (and any "helper" functions which you develop) will constitute a module named "proj11.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 "project11.support.h" (appended below) includes all relevant declarations, along with descriptive comments.

3. The file "project11.driver.o" contains the instructor-supplied driver module.

4. The file "project11.data" contains a sample data set. 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 stubs for the required functions, then translate, link and execute the program to explore the behavior of the driver module.

```
/*****************************************************************/
/*  Declarations for Project #11                                 */
/*****************************************************************/

struct student
{
  unsigned long number;      /* student's ID number (key)      */
  char name[25];             /* student's name                 */
  unsigned short exam1;      /* points on Exam #1              */
  unsigned short exam2;      /* points on Exam #2              */
  unsigned short hw;         /* points on homework             */
  unsigned short total;      /* total points (exams + hw)      */
  float percent;             /* percentage of available points */
};

struct table
{

  unsigned short capacity;   /* number of elements in table    */
  unsigned short count;      /* number of students in table    */
  unsigned short available;  /* total points available         */
  struct student* memory;    /* pointer to array of students   */
};

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

int search( struct table*, unsigned long, struct student** );

/*****************************************************************/
/*  Function:  delete                                            */
/*                                                               */
/*  Purpose:  delete a student from the table, if the            */
/*    student is present in the table.                           */
/*                                                               */
/*  Arguments:                                                   */
/*    pointer to table of students                               */
/*    identification number of student to be deleted             */
/*                                                               */
/*  Return value:                                               */
/*    1 (true) if student deleted, 0 (false) otherwise           */
/*****************************************************************/

int delete( struct table*, unsigned long );
```

```
/*****************************************************************/
/*  Function:  insert                                            */
/*                                                               */
/*  Purpose:  insert a student into the table, as long           */
/*    as there is room in the table and the student is not       */
/*    already present in the table.                              */
/*                                                               */
/*  Arguments:                                                   */
/*    pointer to table of students                               */
/*    identification number of student to be inserted            */
/*    pointer to name of student                                 */
/*    points on Exam #1                                          */
/*    points on Exam #2                                          */
/*    points on homework                                         */
/*                                                               */
/*  Return value:                                                */
/*    1 (true) if student inserted, 0 (false) otherwise          */
/*****************************************************************/

int insert( struct table*, unsigned long, char*, int, int, int );
```