ICOM 4036 – Programming Languages

Programming Assignment 1

Low-Level Programming

Due on Friday February 27

Objectives

- Gain experience with low-level language programming
- Understanding implementation of high-level languages on top of low-level languages
- Apply abstraction design techniques to implement highly generic functions

Problem Statement

For this assignment statement, you will implement a sorting algorithm capable of sorting objects of any sort. You will use the **SPIM** simulator as your development platform. Please consult the course website for documentation pertaining to the simulator. You get to choose the sorting algorithm, but you will be graded proportional to the level of difficulty according to the following scheme:

Algorithm	Percent extra credit
$O(n^2)$ iterative algorithm	0%
O(nlog(n)) iterative algorithm	5%
O(nlog(n)) recursive algorithm	10%

You extra credit will count toward you overall score in programming assignments.

Your sorting algorithm should be implemented as a function that receives four parameters: the address of the array to be sorted, the number of objects to be sorted, the size of each object and the address of a strcmp-type comparison function. The function should sort the objects in the array using the comparison function.

The code in the next figure is C++ version of a function similar to the one that you will implement that uses the insertion sort algorithm. However, the example function can only sort arrays of integers in increasing numerical order. You must generalize the code and then translate it to MIPS assembly language.

```
// Sarray - Selection sort function for one dimension array
void sarray(int larray[],  int nObjects) {
  for (int i = nObjects - 1; i>=0; i--) {
    int max = 0;
    for (int j=1; j<=i; j++) {
      if (larray[j] > larray[max]) {
        max = j;
      }
    }
    int temp = larray[i];
    larray[i] = larray[max];
    larray[max] = temp;
  }
}
```

Part A (50 points) Provide a working version of the sorting algorithm in C++

Part B. (50 points) Provide a working version of the sorting algorithm in MIPS/SPIM assembly language. You must provide the sorting function, an example comparison function and a main subroutine that tests the functionality of the sorting algorithm.

Software Development Platform

You will carry out all your software development using the tools available at the Linux Academic Computing Lab (Amadeus). You may work on your personal Linux-based PC, but you must make sure that your code works with the software development tools available at Amadeus as this is the infrastructure that the staff will use to grade your assignment.

Remember from the prontuario that your programming assignments will be graded according to the following late penalty policy:

Days Late	Percent Deduction
1 day late	25%
2 days late	50%
3 days late	100%

Programming assignments will be graded for both correctness and quality according to the following weights:

Criteria	Weight (%)
Correctness	60%
Design	20%
Efficiency	10%
Style & Documentation	10%