COMP.2030 Homework 3

Consider a linear search program in C on the right.

Assume array, X, contains only positive integers greater than 0 (you can initialize X with arbitrary values in MIPS). The last value in X is set to 0 to mark the end of the array.

In C, we would pass an argument to `search()` to assign the value of 'key', but it can be considered global, as it will be stored in a register to be used within `search()`.

```
void main() {
    int X[];
    int key = 5;
    index = search();
    print("%d", index);
}

int search(){
    int i = 0;
    int found = 0;
    while (!found && X[i] != 0) {
        if (X[i] == key) {
          found = 1;
        } else {
          i++;
        }
    }
    if (found) {
        return i;
    } else {
        return -1;
    }
}
```

1. Write both a C-- and a pseudo-assembly version of the C program on the right (consult homework 2 for details and help on the restrictions of both).

2. Write MIPS code from the pseudo-assembly code in part (1).

3. Implement the MIPS code in part (2) in MARS. You can include an arbitrary array X and initialize the key value in a register.

Review the grading policy for assembly homework (on next page) before beginning part (3).

What to submit:

Include the following documents:

1. Your C-- and pseudo-assembly translation of the code in part (1).
2. The MIPS code in part (3) with comments as a plain text asm file.
3. An explanation, written in your own words, of how you performed each translation, and why each translation step preserves the behavior of the previous version.

Assembly homework will be graded according to the following criteria:

Written explanation (70%), comments (25%), correctness of execution (5%).

- By far, the **most** important result of the exercise is to understand the procedure, the code at each step, and the relationship between the steps along the way. The document you provide to explain your work and *why* it works is the most significant part of the grade. You can get an A on the assignment even if there is a technical error but cannot pass if there is no explanation for why you turned in what you did. So it is better to think through the exercise carefully and thoroughly even if the code is imperfect than to turn in technically correct code you don't understand.

- Comments are also important and will be graded, too.
  - Each function must be preceded by comments, including the name of the function, the arguments used and the registers they are stored in, and how the return value is passed.
  - Each line of MIPS instruction is typically commented on the right in a short C-like statement, so that comments on the right provide enough info on the flow of the function.

- Read the homework assignments carefully. If you miss any of the requirements, points will be taken off (yes, even if it compiles fine).

- As always, no plagiarizing. Remember the university and class policy on academic integrity.

Working with your classmates on homework is fine, if you write your code. Copying someone else's code or letting someone counting your code counts as plagiarizing. The same is true for using AI tools: getting help and advice is OK but copying its output exactly without understanding is not. As such, please make sure that you fully explain all your work in your own words, and that it is completed is written by you and you alone. The logic and reasoning can be similar, but not the explanations itself. While some code can only be written in one way, but the reasoning and explanation will differ between each person. Please write the names of the people you worked with, as well as any tools you may have used (such as ChatGPT, Gemini, or Claude), and what they contributed in your explanation document clear up any confusion.