

# CSE31: Project #2 – Binary Search

## Overview

You may choose to do this project by yourself or with a partner. The objective of this project is to practice the MIPS coding skills you have learned in the class. You will be implementing 2 algorithms in this project.

## Project Details

All the source code is contained in **BinarySearch.s**. This program creates a list of integers based on user inputs. Then it performs **Insertion Sort** on the list. Finally it searches for a user defined key using **Binary Search**. Your task is to implement 3 of the functions described below. **Make sure you DO NOT modify the main method!**

In order to understand how arguments are passed to the following functions, study the main function **CAREFULLY**.

- **printList**: It is a function to print out the content of a list. It takes in **a list** and **its size** as arguments. It does not return any value.
- **inSort**: It performs **Insertion Sort** in **ascending order** on a list. It takes in **a list** and **its size** as arguments. It returns the sorted list as a **new list**. You may use **sorted\_list** defined in the data segment in this function.
- **bSearch**: It performs a **recursive Binary Search** of a key on a list. It takes in **a list, its size, and a search key** as arguments. It returns 1 if the key exists in the list, otherwise it returns 0. You must implement this algorithm **recursively**; therefore, be aware of the use of **stack memory**.

**Create as many test cases as possible so that your program is free of error.**

## Sample output from BinarySearch.s:

### Test case #1:

Enter size of list (between 1 and 25): 5

Enter one list element:

5

Enter one list element:

4

Enter one list element:

7

Enter one list element:

3

Enter one list element:

8

Content of list: 5 4 7 3 8

Content of list: 3 4 5 7 8

Enter a key to search for: 5

Key found!

-- program is finished running --

**Test case #2:**

Enter size of list (between 1 and 25): 8  
Enter one list element:  
8  
Enter one list element:  
7  
Enter one list element:  
6  
Enter one list element:  
5  
Enter one list element:  
4  
Enter one list element:  
3  
Enter one list element:  
2  
Enter one list element:  
1  
Content of list: 8 7 6 5 4 3 2 1  
Content of list: 1 2 3 4 5 6 7 8  
Enter a key to search for: 12  
Key not found!  
-- program is finished running --

**Test case #3:**

Enter size of list (between 1 and 25): 5  
Enter one list element:  
7  
Enter one list element:  
3  
Enter one list element:  
5  
Enter one list element:  
5  
Enter one list element:  
2  
Content of list: 7 3 5 5 2  
Content of list: 2 3 5 5 7  
Enter a key to search for: 5  
Key found!  
-- program is finished running --

## What to hand in

When you are done with this project, you are ready to submit your work. Make sure you have included the following before you press Submit:

- Your completed **BinarySearch.s**
  - A text document containing at least **5 test cases** (see examples above)
  - In order for your assignment to be graded, you must **demo** your code to your instructor **AFTER** submission.
-