

Overview

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.

Description of wordsearch.c

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 the three (3) functions described below. **Make sure you DO NOT modify the main method!**

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

- **printList**: This function prints the contents of a list. It takes in a list and its size as arguments. It does not return any value.
- **inSort**: This function 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**: This function 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 Runs (user input shown in blue, with each run separated by a dashed line)

```
-----SAMPLE RUN 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 --

-----SAMPLE RUN 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 --
```

```
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 --
```

Collaboration

You must credit anyone you worked with in any of the following three different ways:

1. Given help to
2. Gotten help from
3. Collaborated with and worked together

What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

Before you submit, make sure you have done the following:

- Your code compiles and runs on a Linux machine.
- Attached **BinarySearch.s**.
- Attached a text document named **testRuns.txt** containing at least 5 test cases (see examples above).
- Filled in your collaborator's name (if any) in the "Comments..." textbox at the submission page.

Also, remember to demonstrate your code to the TA or instructor before the deadline.