Homework 4: High-Low Game

Released Friday 9/16/16 - 5:00pm

Due Friday 9/23/16 - 11:59pm

Goals:

- Learn how to implement Binary Search algorithm
- Be able to implement a simple high-low guessing game using Binary Search.

Description:

The High-Low Game is where one player thinks of a number, between **1** and **100** (inclusive), and then the other player tries to guess the number by repeatedly asking if the number is larger than certain number. The computer plays the role of the player that guesses the number.

The goal is to write a program that will guess the user's number with the fewest number of questions in worst case scenario.

The algorithm you are asked to implement is a common algorithm for searching on *sorted* data, and is known as **Binary Search**. It has the nice property that it only takes log(n) guesses in the worst case to determine if an entry is present, as opposed to n guesses for a more naive search that examines each element.

Prerequisites:

- Understanding of Binary Search algorithm.

Given:

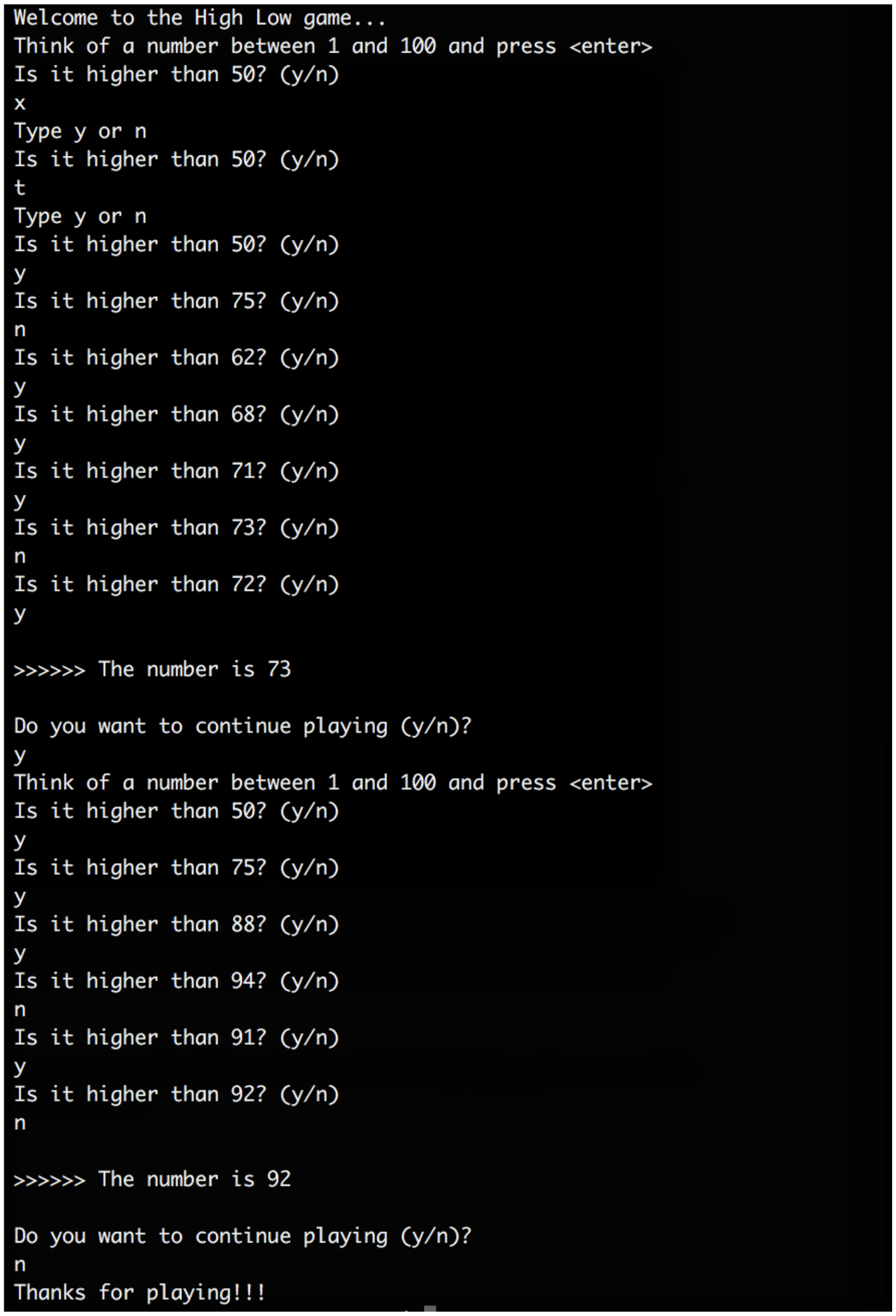
- **NONE -** You will start from Scratch

Steps:

- The program should print "**Welcome to the High Low game...**" on it's own line when the program is first run.
- Each time the player starts a new run of the game, the program must ask the player "**Think of a number between 1 and 100 and press <enter>**".
- Then ask "**Is it higher than <NUMBER>? (y/n)**".
- Take in the answer is '**y**' or '**n**'
- The correctly guessed number will be displayed as "**>>>>>> The number is <number>**"
- After each game, the program should ask, "**Do you want to continue playing (y/n)?**"
- Continue playing the game as long as the user inputs '**y**', otherwise print "**Thanks for playing!!!**" before exiting the program.

Demo

An **example run** of the program is shown below:



```
Welcome to the High Low game...
Think of a number between 1 and 100 and press <enter>
Is it higher than 50? (y/n)
x
Type y or n
Is it higher than 50? (y/n)
t
Type y or n
Is it higher than 50? (y/n)
y
Is it higher than 75? (y/n)
n
Is it higher than 62? (y/n)
y
Is it higher than 68? (y/n)
y
Is it higher than 71? (y/n)
y
Is it higher than 73? (y/n)
n
Is it higher than 72? (y/n)
y

>>>>>> The number is 73

Do you want to continue playing (y/n)?
y
Think of a number between 1 and 100 and press <enter>
Is it higher than 50? (y/n)
y
Is it higher than 75? (y/n)
y
Is it higher than 88? (y/n)
y
Is it higher than 94? (y/n)
n
Is it higher than 91? (y/n)
y
Is it higher than 92? (y/n)
n

>>>>>> The number is 92

Do you want to continue playing (y/n)?
n
Thanks for playing!!!
```

Submission:

These are the files that need to be in the Vocareum "work" directory for this assignment.

- **high-low.c**

Extra Credit Question:

Introduction:

Assume there is a sorted array [2,5,8,9,11,14]. Now imagine we push the array from its tail. If we push once, the first one number would be moved to the end of the array. The array becomes an anomaly sorted array [5,8,9,11,14,2]. If we push twice, the first two numbers would be moved to the end of the array. The anomaly sorted array would be [8,9,11,14,2,5]. If we push three times, the array would become [9,11,14,2,5,8]. We are allowed to push *n* times, where $0 \le n \le \textit{length of array}$.

You may find that if $n = 0$ or $n = \textit{length of array}$, the anomaly sorted array after push would be the same as original one. However, if *n* is other values, even though the whole array would not be sorted, it consists of two sorted subarrays. That is why we call it anomaly sorted array.

Question:

Given an anomaly sorted array from pushing a regular sorted array unknown times and a target number, find and return the index of the target number if it exists in the anomaly sorted array. Otherwise return -1.

Given:

- binarySearch.c

Example:

Anomaly sorted array 1 = [4,7,9,10,0,2,3]
If target number = 9, return 2, which is the index of 9 in the array.

Anomaly sorted array 2 = [8,10,12,15,2,7]
If target number = 2, return 4, which is the index of 2 in the array.

Anomaly sorted array 3 = [12,14,20,22,4,6,9,10]
If target number = 11, return -1, because 11 does not exist in the array.

Note: You are supposed to use *binary search* algorithm to solve this question. Scan the whole array once to find target will not be given extra credit.

Submission:

These are the files that need to be in the Vocareum “work” directory for this assignment.

- **binarySearch.c**