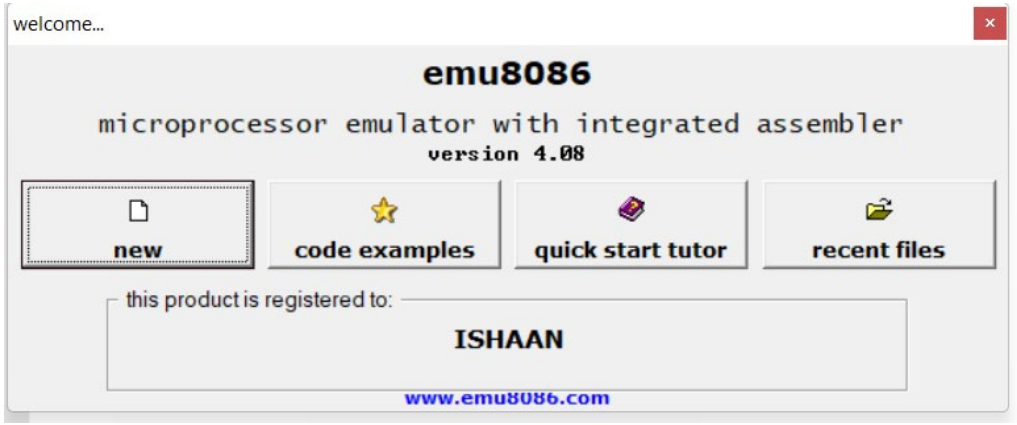
8086

Binary Search
https://github.com/ahmedokka29/microprocessor8086-project

Introduction

- In our project we use the x86 assembly language to build a Binary search algorithm in which we used a lot of what we have learned in our microprocessor course and from our own research.
- We used emu8086 to assemble and run our program.



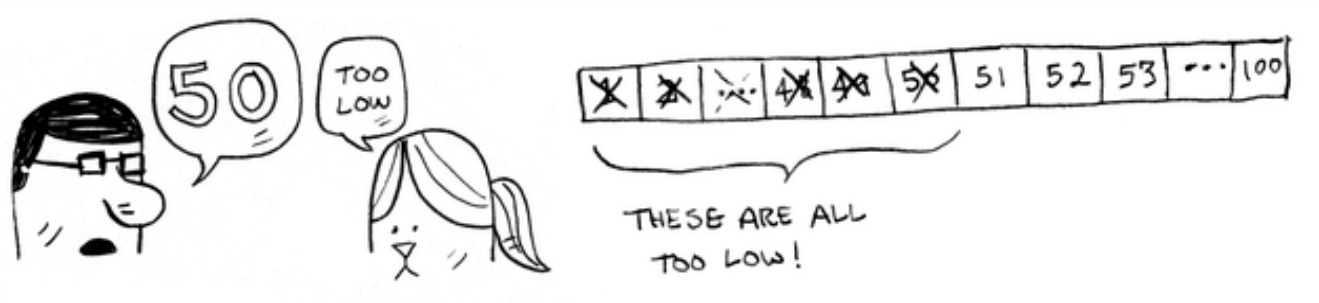
Binary Search

- Binary search is an algorithm; its input is a sorted list of elements. If an element you're looking for is in that list, binary search returns the position where it's located. Otherwise, binary search returns null.

Binary Search

- Here's an example of how binary search works. I'm thinking of a number between 1 and 100.

Start with 50.

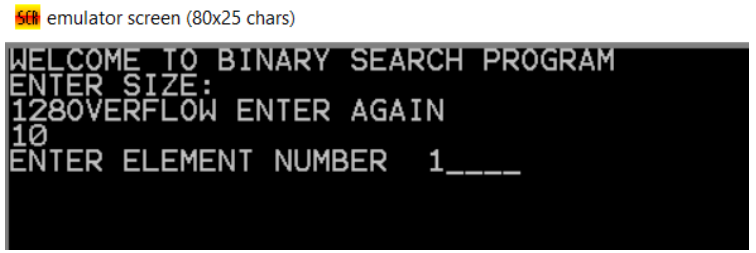


Too low, but you just eliminated *half* the numbers! Now you know that 1–50 are all too low. Next guess: 75.

Too high, but again you cut down half the remaining numbers! *With binary search, you guess the middle number and eliminate half the remaining numbers every time*. Next is 63 (halfway between 50 and 75).

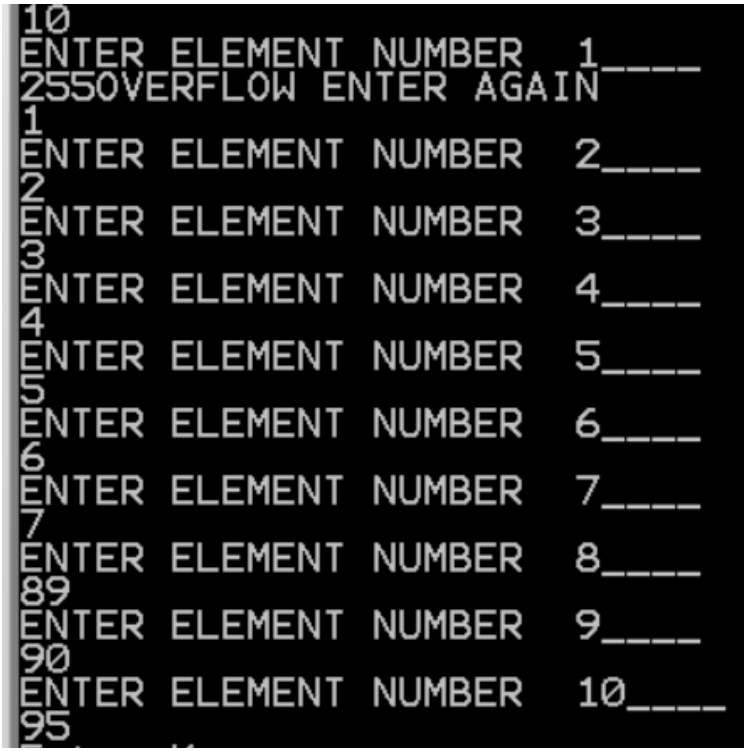
How our program work #1

- Firstly, our program can save and array up to 127 elements.
- When the user enters size greater than 127, the program shows error message “overflow try again” which gives the opportunity to user to enter right value.



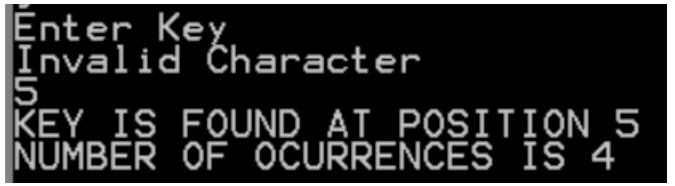
How our program work #2

- The user should fill the array list sorted within certain range 0 to 127.
- Case1: If the user enters number outside that range the program will display error message and let the user enter the number again.
- Case2: If the user enters number lower than the last previous number that's not allowed, and the program displays that number is not sorted and let the user sorted number again.



How our program work #3

- After filling the sorted list, the program let the user search on certain key to know its position and how much its occurrences.
- The program displays the first position that the key appears at.



Team Members

Sayed Hassan

- https://github.com/9mm-bot

Ziad Mohamed

- https://github.com/ZiadSENG

Ibrahim Mohamed

- https://github.com/hemagazzar

Ahmed Tarek

- https://github.com/ahmedashour28

Ahmed Okka

- https://github.com/ahmedokka29

Thanks for your attention