I. Purpose of the experiment

For this assignment, you will learn about functional programing. After learning it, you will advocate that real-world programmers should pay attention to/learn functional programming because it helps solve real-world problems/is valid in current trends in the workplace. Then you should complete the given tasks by using functional programing language.

II. Experimental content

1. Advocates that real-world programmers should pay attention to/learn functional programming because it helps solve real-world problems/is valid in current trends in the workplace.
2. Writes a small program in a functional programming language to solve one of the following problems:
    a. Find the K^{th} element of a list. (By convention, the first element in the list is element 0)

Or

    b. Determine whether a given integer number is prime.

Functional languages: Scala， Racket，Scheme，Haskell，LISP，Elixir，F#

III. Experimental steps

Step1:

Learn about the functional programing from given articles and sources on the web. Then briefly describe functional programing.

Step2:

Summarize the advantages of functional programing and advocate why programmers should learn functional programming.

Step3:

Use Lisp language to complete problems. Briefly explain the implementation steps and the reasons for choosing the language.

Step4:

Make a analysis and conclusion.

IV. Experimental results and analysis(Discussion)

1. What is the task and what is functional programing?

The task consists of two parts. The first part is advocate programmers to use functional programming language. The second part is using one of functional programming languages to find K^{th} element of a list or determine whether a given integer number is prime.

Functional programming is a non-von Neumann programming language. Its fundamental operation is the application of functions to arguments. A main program itself is written as a function that receives the program's input as its argument and delivers the program's output as its result.

2. Why should learn functional programming?

(1) Functional programs contain no assignment statements, so variables, once given a value, never change. This has the nice advantage of preventing side effects and race conditions; making concurrent programming much much easier.

(2) Expressions can be freely replaced with variables and (more importantly in my opinion) vice versa.

(3) Functional programs are more modular, and functionality is easier to swap.

(4) Easy to maintain. It takes advantage of techniques like higher order functions, functional composition, and currying helps make code more reusable. And code reuse can result in fewer lines of code. Additionally, because these techniques allow code to be written at a higher level of abstraction, this tends to make it easier to understand what the code is doing, therefore improving maintainability.

(5) It has good concurrency, which is becoming more and more important in today's software development

3. Complete the problem by using functional programming language

I decide to use LISP. Because it has an online compilation website, no need to configure the environment and it is an old programming language with a simpler syntax.

Then I complete all of two problems because I want to practice more about this programing language.

For the frist problem:

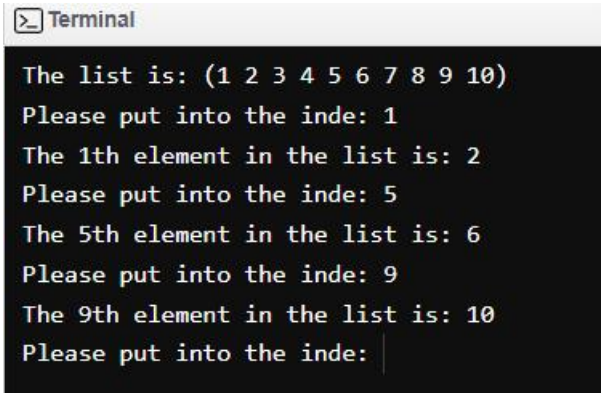
Step1: Create a list variable and store ten numbers

Step2: Read the index entered by the user

Step3: Use the loop body to traverse the list

Step4: Write a function that does the lookup, and calls it on each traversal

Step5: Print the result

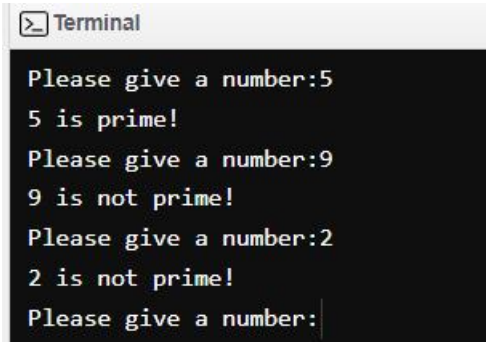


Graph 1 Result of problem1

For the second problem:

Step1: Read the number entered by user

Step2:Write a function that check if a number is a prime and call it. In this function, use loop body combined with square root operation to check.



Graph 2 Result of problem2

V. Experimental experience and gains

1. Gain an in-depth look at functional programming languages.
2. Learned the advantages of functional programming languages and their advantages, trends in software development today.
3. Learn about the role of parallelism in software development.
4. Understand how functional programming languages differ from other languages.
5. Practice with functional programming languages.

VI. Core Code

Problem1:

```
(defun getElement(index tempList)
    "Search nth element in a list"
    (if(or(> index (length tempList))(< index 0))
        (format t "~% Index out of bound!")
        (format t "~% The ~Dth element in the list is: ~D" index (nth index tempList))
        )
)

(defun main()
    "The main function"
    (setq tempList (list 1 2 3 4 5 6 7 8 9 10))
    (terpri)
    (princ "The list is: ")
    (write tempList)
```

```
        (loop
            (terpri)
            (princ "Please put into the inde: ")
            (setq x(read))
            (getElement x tempList)
        )
    )

    (main)

    Problem2:
    (defun judgePrime(num)
        "Determine whether the given integer number is prime"
        (if(and(loop for i from 2 to (sqrt num) never (= 0 (mod num i)))(> num
2))
            (format t "~%~d is prime!" num)
            (format t "~%~d is not prime!" num)
        )
    )

    (defun main()
        "The main function"
        (terpri)
        (loop
            (terpri)
            (princ "Please give a number:")
            (setq num(read))
            (judgePrime num)
        )
    )

    (main)
```

VII. Reference

1. https://www.tutorialspoint.com/lisp/lisp_operators.htm

2. https://blog.csdn.net/studyhard232/article/details/69664695
3. https://blog.csdn.net/lffup/article/details/124540319
4. https://www.codenong.com/2036244/
5. https://allampersandall.blogspot.com/2012/01/functional-programming-matters.html
6. https://dev.to/allanmacgregor/you-should-learn-functional-programming-in-2018-4nff