**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**\*\*\*\*\*\*\*\*\***

****

**PROJECT REPORT**

**ASSEMBLY LANGUAGE AND COMPUTER ARCHITECTURE LAB**

**Class: 122158**

**Name: Nguyễn Thị Nga**

**Student ID: 20184296**

**Group 10**

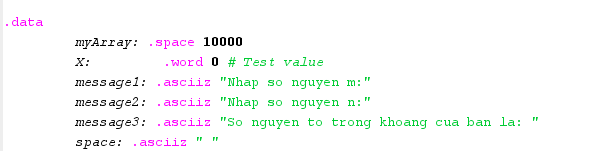
*HaNoi, June 25 2021*

1. **Project 1.2**

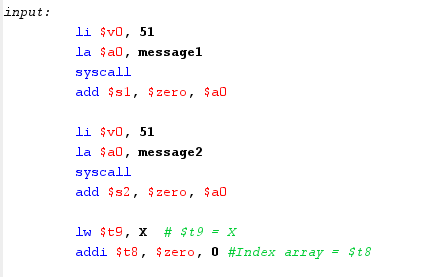
***- Find all prime numbers (such as 2, 3, 5, 7...) in a range from the integer N to the integer M.***

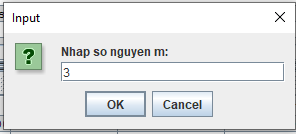
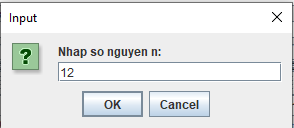
* **Implementation**:
* Step 1: Initialize an array myArray to store the result after running the program and an integer X = 0 to check if it is prime. ( If X= 0 then it is prime, otherwise it is not prime.)
* Step 2: Get data (m,n) from the user input from the keyboad, the output will be all prime numbers in the range of those 2 integers.
* Step 3: Use a loop from m to n, let it divide from 2 to itself -1, if any number is divisible, assign th evalue X=1 and check the next number, otherwise X remains at its original value.
* Check if X = 0, then we will add to the original myArray and after the loop is over we will print out the result.
* **Perform:**

1. *Declare:*



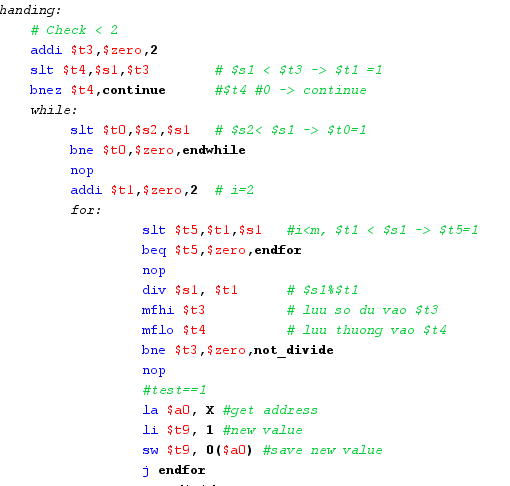
1. *Input:*



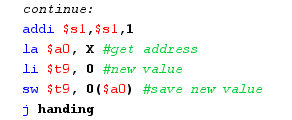


* The value of 2 user-entered integers will be stored in 2 registers $s1, $s2.
* Assign register $t9 initial check value = 0, array address (myArray) $t8

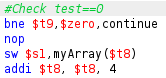
1. *Handler function:*



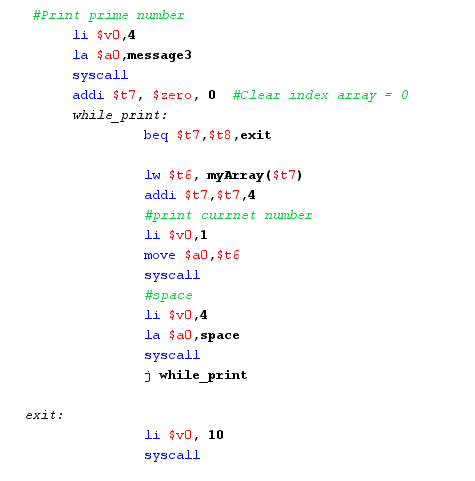
* If it is prime, then assign the check register $t9 = 1
* We use register $t4 to check if the input number is < 2, if it is smaller, we will jump to continue to increase the value to 1, prepare to check the next number



* Check if register $t9 = 0, then we save that prime number into array array, increment the array address by 4, otherwise, we will jump to continue to check the next number



1. *Output*



* We use register $t7 to assign the array address back to the original then use a while loop to get the result. The loop stops recording the value of 2 registers $t7 and $t8 (the address of the original declared array) are equal.
* The result is: All prime number from 3 to 12 is:



1. **Project 2.1 Curiosity Marsbot**

Xe tự hành Curioisity Marsbot chạy trên sao Hỏa, được vận hành từ xa bởi các lập trình viên trên Trái Đất. Bằng cách gửi đi các mã điều khiển từ một bàn phím ma trận, lập trình viên điều khiển quá trình di chuyển của Marbot như sau:

|  |  |
| --- | --- |
| **Mã điều khiển** | **Ý nghĩa** |
| 1b4 | Marbot bắt đầu chuyển động |
| c68 | Marbot đứng im |
| 444 | Rẽ trái 90 độ so với phương chuyển động gần nhất và giữ hướng mới |
| 666 | Rẽ phải 90 độ so vơi phương chuyển động gần nhất và giữ hướng mới |
| dad | Bắt đầu để lại vết trên đường |
| cbc | Chấm dứt để lại vết trên đường |
| 999 | Tự động quay trở lại theo lộ trình ngược lại. Không vẽ vết, không nhận mã khác cho tới khi kết thúc lộ trình ngược.  Mô tả: Marsbot được lập trình để nhớ lại toàn bộ lịch sử các mã điều khiển và khoảng thời gian giữa các lần đổi mã. Vì vậy, nó có thể đảo ngược lại lộ trình để quay về điểm xuất phát. |

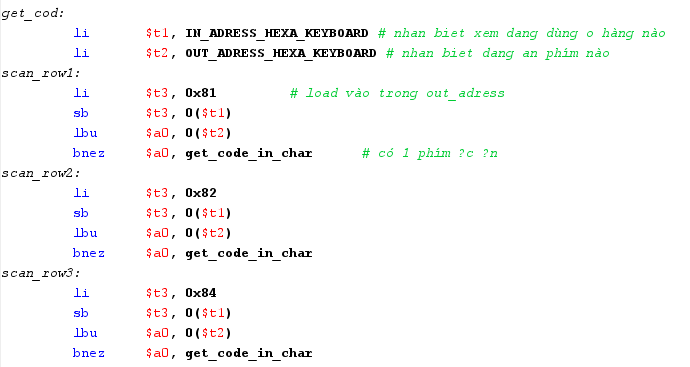
Sau khi nhận mã điều khiển, Curioisity Marsbot sẽ không xử lý ngay, mà phải đợi lệnh kích hoạt mã từ bàn phím Keyboard & Display MMIO Simulator. Có 2 lệnh như vậy:

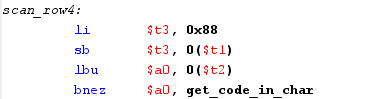
|  |  |
| --- | --- |
| **Kích hoạt mã** | **Ý nghĩa** |
| Phím Enter | Kết thúc nhập mã và yêu cầu Marbot thực thi. |
| Phím Del | Xóa toàn bộ mã điều khiển đang nhập. |

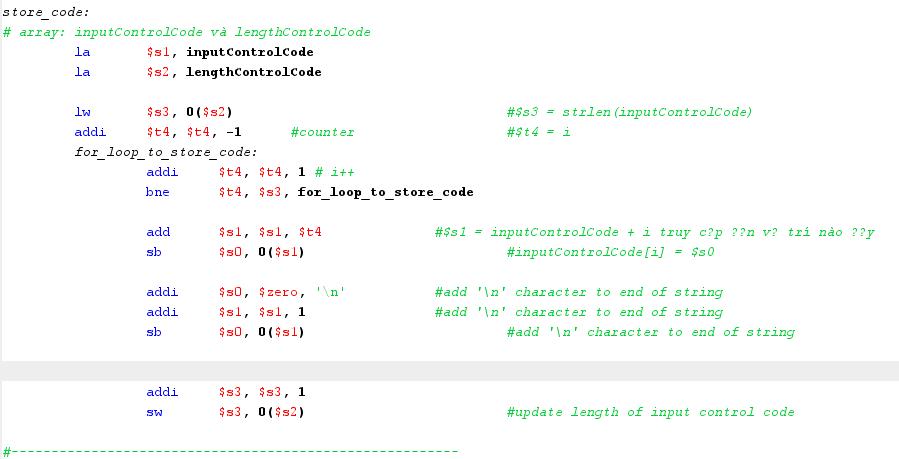
Hãy lập trình để Marsbot có thể hoạt động như đã mô tả.

Đồng thời bổ sung thêm tính năng: mỗi khi gửi một mã điều khiển cho Marsbot, hiển thị mã đó lên màn hình console để người xem có thể giám sát lộ trình của xe.

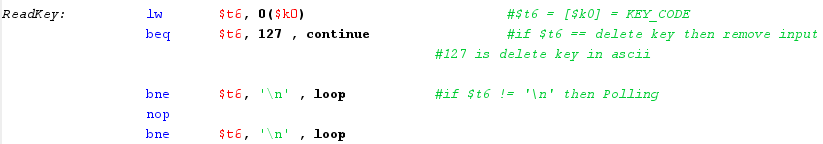
* **Implementation**:
* **Step 1:** Input command from Digital Lab Sim
* **Step 2:** Input “Enter” from Keyboard Display to confirm command from step 1 or “Delete” to delete all command which were entered.
* **Step 3:** Marbot will go base on command we input from digital lab sim
* **Declare:**
* **Input**

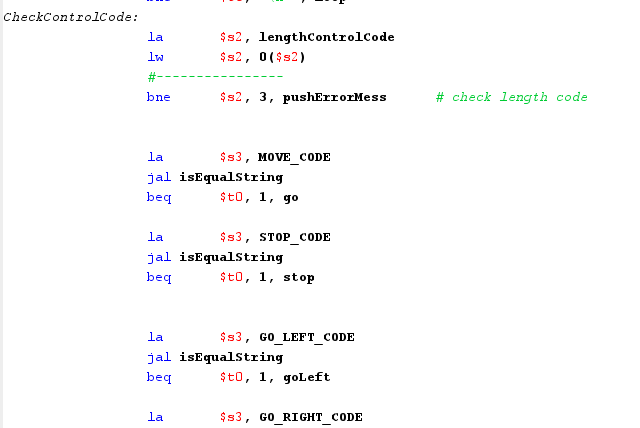
****

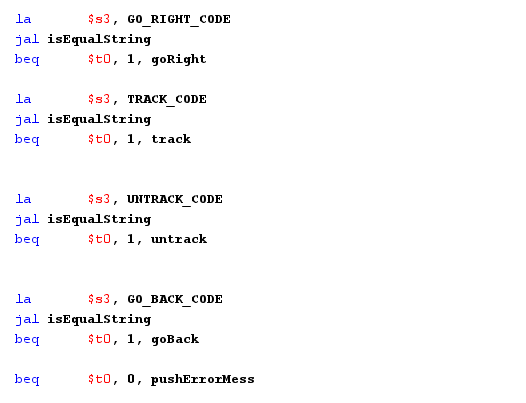
****

****

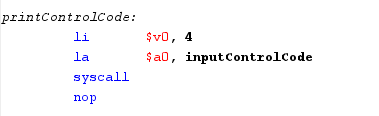
* **Check code**

****

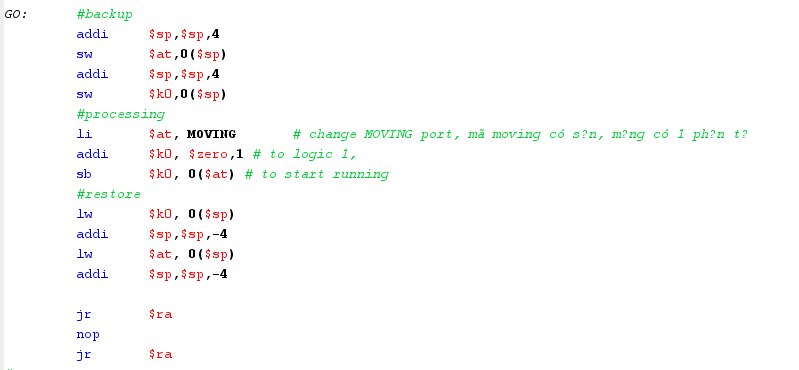


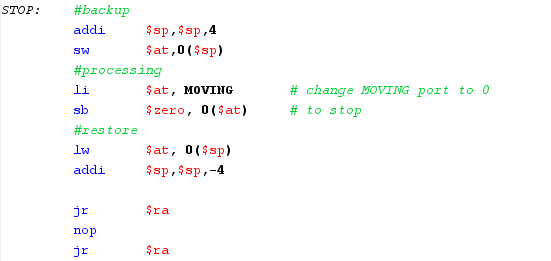


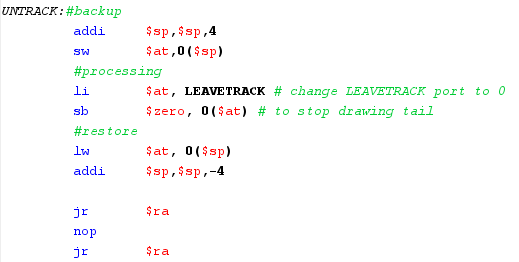
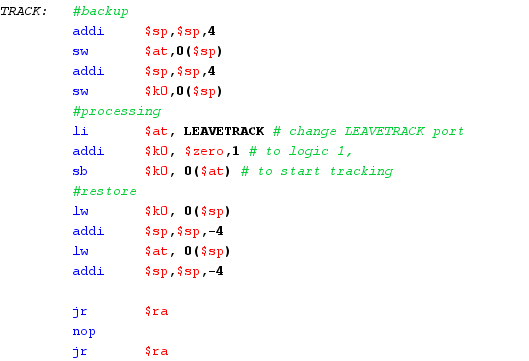
* **Print code to console**



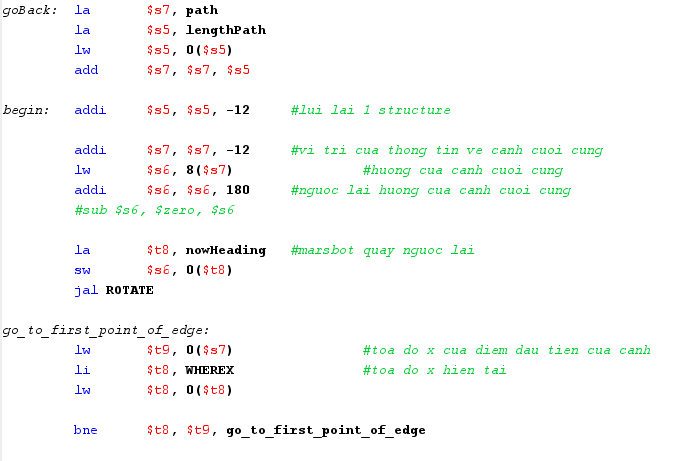
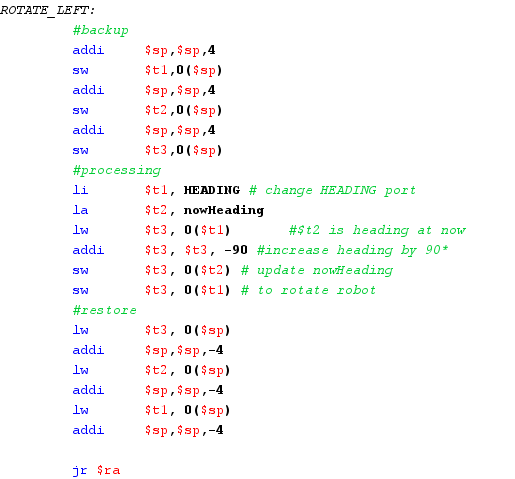
* Control command to Marbot run











* **Demonstration**