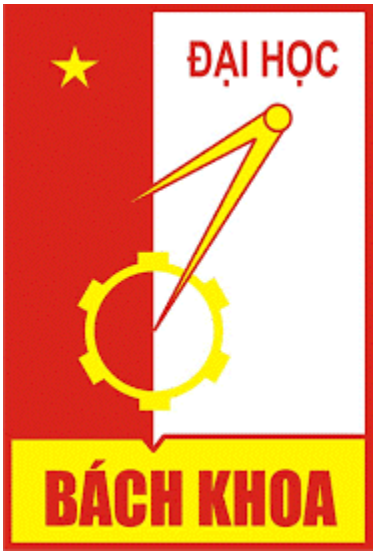
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**
**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

---




# Mini-projects
# Computer architecture

**GVHD: TS. Lê Xuân Thành**
**Nhóm: 1**
**Thành Viên: Mai Mạnh Thục – 20176885**
**Lê Hoàng Anh Trung – 20176892**
**Môn học: Thực hành Kiến Trúc Máy Tính**
**Mã Lớp: 113835**

# Project 1:

Create a program to input a text line from the keyboard and test if it is a palindrome. For example: “abc121cba” is a palindrome. Store all palindromes which the user typed into the memory, to make sure that the user doesn’t duplicate palindromes.

Input: “abc121cba”

Output: true

**Phân tích đề bài:**

- Input là 1 xâu kí tự
- Output là lưu toàn bộ các xâu là palindrome mà người dùng đã nhập, đồng thời kiểm tra để đảm bảo không bị trùng lặp palindromes lưu trong bộ nhớ

**Phân tích cách thực hiện:**

- **Ý tưởng thực hiện:**

- Nhập một xâu từ bàn phím, duyệt xâu để kiểm tra điều kiện: đã xuất hiện chưa,ký tự đặc biệt, xâu quá dài, …
- Dùng một thanh ghi để duyệt xâu từ đầu, dùng thanh ghi khác duyệt ngược lại, so sánh 2 thanh ghi
- Lưu xâu palindrome đã được phát hiện vào một mảng

- **Các bước thực hiện**:

- Nhập vào xâu từ bàn phím lưu tạm nó vào bộ nhớ
- Dùng vòng lặp đếm số kí tự của nó xem có vượt quá hay không nếu không nhập lại
- Kiểm tra xem có trùng lặp palindromes hay không bằng cách:
    - Kiểm tra xem đã có palindrome nào được lưu hay chưa
    - Dùng biến tạm trỏ đến địa chỉ của từng palindrome trong bộ nhớ
    - So sánh từ kí tự của xâu vừa nhập với từng palindrome đã lưu trong bộ nhớ
- Kiểm tra tính hợp lệ của xâu bằng cách:
    - Kiểm tra mã ASCII của từng kí tự trong xâu xem có thuộc khoảng giá trị thỏa mãn yêu cầu không
    - [48, 57] : chữ số
    - [65, 90] : chữ cái viết hoa
    - [97, 122] : chữ cái viết thường
    - Viết hoa các chữ cái thường và lưu vào trong 1 vùng nhớ khác để đem đi so sánh ở bước tiếp theo
- Sau khi thỏa mãn các điều kiện trên tiến hành kiểm tra xem xâu có phải là 1 palindrome hay không:
    - Dùng 1 thanh ghi để duyệt từ cuối xâu, dùng 1 thanh ghi khác để duyệt từ đầu xâu.

- Kiểm tra vị trí duyệt xem đã gặp nhau hay chưa.
- Khi có một vị trí có giá trị khác nhau thì kết thúc duyệt, xâu đó không phải palindrome.
- Nếu đã gặp nhau mà không có kí tự nào khác nhau thì chứng tỏ xâu đó là 1 palindrome.
- Lưu palindrome lại trong bộ nhớ.
- Thuật toán:

```
Boolean isPalindrome(String input[]){
  for(i = 0; i<length/2; i++){
    if(input[i] == input[length-i-1]) continue;
    else return false;
  }
  return true;
}
```

- Hỏi yêu cầu người dung muốn nhập tiếp hay không, nếu không thì reset lại phần bộ nhớ không nhằm mục đích lưu trữ palindromes.

**Ý nghĩa của các thanh ghi được sử dụng:**

- $a0: Lưu địa chỉ xâu người dùng vừa nhập vào, và dung cho 1 số thông báo khi sử dụng syscall.
- $v0: Phục vụ sử dụng lệnh syscall
- $a1: Phục vụ sử dụng lệnh syscall
- $a3: Địa chỉ của mảng các palindromes trong bộ nhớ
- $s0: Lưu số lượng các palindrome đã lưu trong bộ nhớ
- $t0, $t1, $t2, $t3, $t4, $t5, $t6, $t7, $t8, $t9: Biến tạm dùng khi so sánh

**Chương trình con nếu có:**

- testLocation: kiểm tra vị trí duyệt xâu hiện tại

**Mã nguồn:**

```
# Project giua ky mon THUC HANH KIEN TRUC MAY TINH

.data
	userInput: .space 50	# luu xau vua nhap vao
	userInputx: .space 50	# Xau da qua xu ly de khong phan biet chu hoa chu thuong
	palindromes: .space 1000
	isPali: .asciiz " is a palindrome :)\n\n"
	notPali: .asciiz " is not a palindrome :(\n\n"
	goAgainMessage: .asciiz "Would you like to enter another? (yes/no): "
	goAgainInput: .space 4
```

```
      exitMessage: .asciiz "Program is finished. "
      startPrompt: .asciiz "\nEnter a string to check if it is a palindrome: "
      duplicateString: .asciiz "\nThe String just have entered is exist in
Palindromes\n\n"
      invalid_length: .asciiz "\nLength of string is too long\n\n"
      invalid_char: .asciiz "\nThe string you just entered contains a special
character\n\n"

.text
la $a3, palindromes             # $a3 la dia chi cua mang cac palindrome
main:
    la $a0, startPrompt         # In ra thong bao nguoi dung nhap xau
    li $v0, 4
    syscall
    la $a0, userInput           # dia chi cua xau nguoi dung vua nhap vao
    li $a1, 50                  # So ki tu toi da co the doc vao cua moi lan nhap xau
    li $v0, 8
    syscall                     # doc vao xau nguoi dung nhap

#---------------------------------------------------------------------------------------
# check_lengthInput      Kiem tra xem xau nhap vao co vuot qua do dai cho phep hay khong?
# Su dung vong lap while duyet xau, dem so luong ki tu trong xau
#---------------------------------------------------------------------------------------

check_lengthInput:
    li $t0, 0
    countLength:
     lb $t1, ($a0)
     beqz $t1, end_countLength
     addi $t0, $t0, 1
     addi $a0, $a0, 1
     j countLength
    end_countLength:
    bgt $t0, 26, invalid_lengthInput
    j end_check_lengthInput
    invalid_lengthInput:
     la $a0, invalid_length
     li $v0, 4
     syscall
     j goAgainX
end_check_lengthInput:
    addi $a0, $a0, -1
    sb $zero, ($a0)             # xoa ki tu '\n' cuoi xau vua nhap vao

#---------------------------------------------------------------------------------------
# duplicate : Kiem tra xem xau da ton tai trong bo nho hay khong ?
# Su dung vong lap while, su dung 1 thanh ghi duyet tu dau xau va 1 thanh ghi duyet tu
cuoi xau
# next_checkdup : kiem tra vi tri duyet xau hien tai
# isDuplicate : xau vua nhap da ton tai va tiep tuc hoi nguoi dung muon nhap tiep hay
khong
```

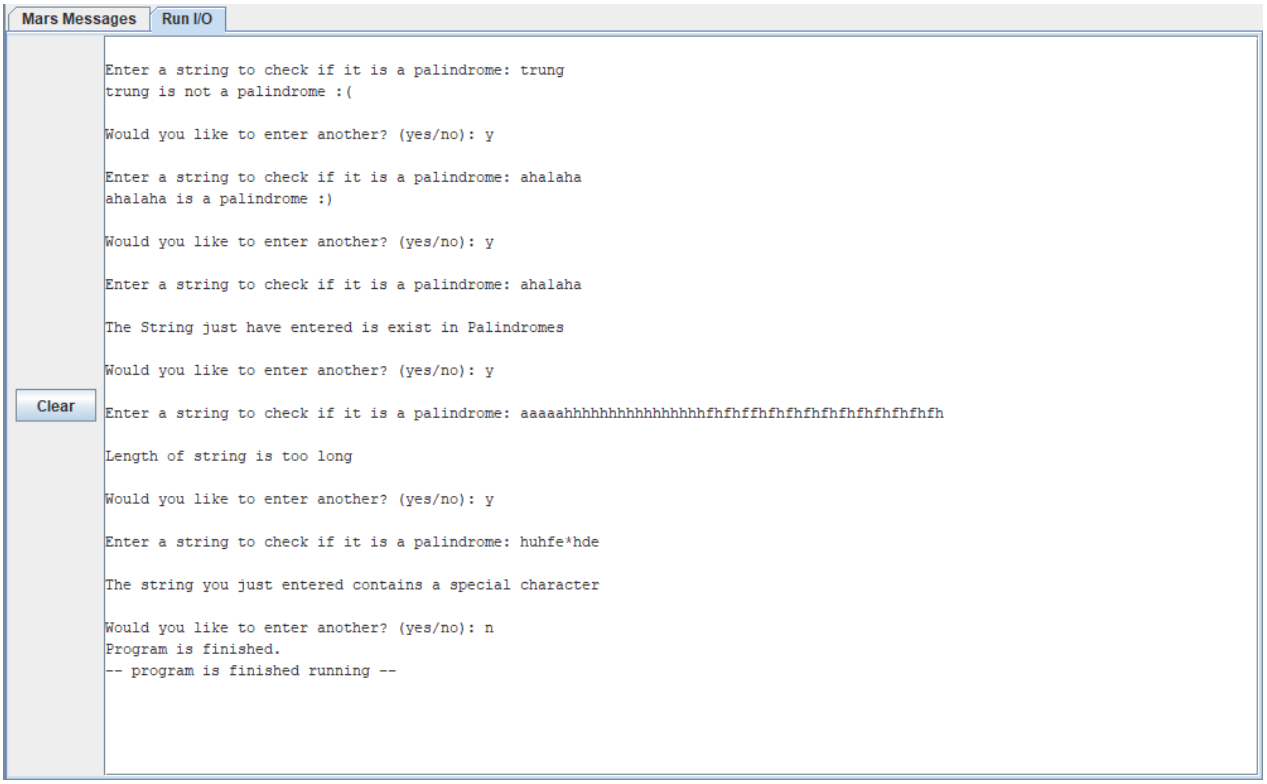
```
# notDuplicate : xau vua nhap chua ton tai trong bo nho va tiep tuc kiem tra tinh hop le
cua cac ki tu trong xau
#-------------------------------------------------------------------------------------

duplicate:
 la $a0, userInput
 la $a1, palindromes
 li $t0, 27
 add $t6, $s0, $zero
 loop_dup:
       beqz $s0, end_duplicate          # Neu palindromes rong -> ket thuc duplicate
       mult $t0, $t6
       mflo $t2
       add $t1, $a1, $t2                # $t1 = dia chi tung palindrome trong bo nho
       sub $t1, $t1 , 27
       add $t5, $a0, $zero              # $t5 = 0(userInput)

       # So sanh xau vua nhap vao voi cac xau da luu trong bo nho (neu co)
       while:
              lb $t2, ($t1)             # $t2 tro den palindromes trong bo nho
              lb $t3, ($t5)             # $t3 tro den xau vua nhap vao dem di so sanh
              beq $t2, $t3, next_check_dup     # palidrome[i] = userInput[i]
              j notDuplicate
       next_check_dup:
              addi $t1, $t1, 1          # $t1 = 1(palindromes[i])
              addi $t5, $t5, 1          # $t5 = 1(userInput)
              lb $t2, ($t1)
              lb $t3, ($t5)
              add $t4, $t2, $t3
              beqz $t4, isDuplicate
              j while
       end_while:
 end_loop_dup:
 isDuplicate:
       la $a0, duplicateString          # In thong bao xau da trung lap trong bo nho
       li $v0, 4
       syscall
       j goAgainX
 notDuplicate:                          # Thong lap thi tiep tuc thuc hien chuong trinh
       addi $t6, $t6, -1
       beqz $t6, end_duplicate
       j loop_dup
end_duplicate:

#-------------------------------------------------------------------------------------
# checkInput :  Kiem tra tinh hop le cac ki tu cua xau vua nhap vao (co ki tu dac biet
khong?)
# Dung vong lap kiem tra tung ki tu trong xau vua nhap co thuoc dung khoang gia tri ma
Ascii cho chu cai va chu so hay khong
#-------------------------------------------------------------------------------------

checkInput:
```

```
    la $a0, userInput
    la $t1, userInputx
    loop:
        lb $t7, ($a0)                       # load a[i] to $t7
        beq $t7, 0, checkPali               # ket thuc checkInput -> checkPali
        bgt $t7, 47, test1
        j invalidChar
        test1:
         blt $t7, 58, addToInputx           # a[0] là 1 chu so tu 0->9, ma ascii [48,57]
         j test2
        test2:
         bgt $t7, 64, test3
         j invalidChar
        test3:
         blt $t7, 91, addToInputx           # a[i]: chu cai viet hoa ma ascii [65,90]
         bgt $t7, 96, test4
         j invalidChar
        test4:
         blt $t7, 123, addToInputx          # a[i]:chu cai viet thuong ma ascii [97,122]
         j invalidChar
    addToInputx:
        bgt $t7, 96, makeCap
        j notCap
        makeCap:
         addi $t7, $t7, -32                 # chuyen chu viet thuong -> viet hoa(a->A)
        notCap:
         sb $t7, ($t1)                      # userInputX[i] = userInput[i]
         addi $a0, $a0, 1
         addi $t1, $t1, 1
         j loop

    # Thong bao la xau co ki tu dac biet va hoi nguoi dung co muon nhap tiep khong.
    invalidChar:
       la $a0, invalid_char
       la $v0, 4
       syscall
       j goAgainX

#-------------------------------------------------------------------------------------------
# checkPali : kiem tra xau vua nhap co phai la 1 palindrome hay khong?
# loop2 : load tung ki tu de kiem tra tinh doi xung
# testLocation : Kiem tra vi tri cac con tro dung de duyet xau neu 2 con tro gap nhau thi
no la 1 palindrome
# isPaliX : thuc hien luu xau palindrome va tang so luong phan tu trong mang palindromes
len 1
# notPaliX : In thong bao va hoi nguoi dung co muon nhap tiep hay khong
#-------------------------------------------------------------------------------------------

checkPali:

    la $t4, userInputx                  # $t4 = dia chi dau xau vua nhap (userInput[0])
    addi $t1, $t1, -1                   # $t1 = dia chi cuoi xau vua nhap (userInput[n])
```

```
loop2:
    lb $t3, ($t4)                  # $t3 = userInput[0]
    lb $t2, ($t1)                  # $t2 = userInput[n]
    beq $t3, $t2, next             # userInput[0] = userInput[n]
    j notPaliX                     # khong la 1 palindrome
    next:
     jal testLocation
     nop
     addi $t4, $t4, 1
     addi $t1, $t1, -1
     j loop2

testLocation:
    beq $t4, $t1, isPaliX          # 2 con tro gap duyet xau gap nhau thi ket thuc
    addi $t1, $t1, -1              # Kiem tra xem co tiep tuc duyet them hay khong
    beq $t4, $t1, isPaliX
    addi $t1, $t1, 1
    jr $ra

isPaliX:
    la $a0, userInput
    li $v0, 4                      # In ra thong bao xau vua nhap la 1 palindromes
    syscall
    la $a0, isPali
    syscall
    la $a0, userInput
   li $t8, 27
    loop3:                         # Luu xau vao mang palindromes da khai bao
     lb $t5, ($a0)
     beqz $t8, end_loop3
     sb $t5, ($a3)
     addi $a0, $a0, 1
     addi $a3, $a3, 1
     addi $t8, $t8, -1
     j loop3
    end_loop3:

    addi $s0, $s0, 1               # Luu lai so luong palindromes da luu trong bo nho
    j goAgainX

notPaliX:                          # Thuc hien khi xau khong phai 1 palindromes
    la $a0, userInput
    li $v0, 4
    syscall
    la $a0, notPali
    syscall                        # In ra thong bao xau khong phai la 1 palidrome
    j goAgainX

goAgainX:                          # Hoi nguoi dung xem co muon nhap tiep hay khong?
    la $a0, goAgainMessage
    syscall
    li $a1, 4
```

```
        li $v0, 8
        la $a0, goAgainInput
        syscall
        lb $t0, ($a0)
        beq $t0, 121, goAgain          # input[0] = 'y'
        j exit
#-------------------------------------------------------------------------------------------
# Ket thuc chuong trinh
# reset lai phan bo nho khong phuc vu luu tru palindromes
#-------------------------------------------------------------------------------------------

exit:
    la $a0, exitMessage
    li $v0, 4
    syscall
    la $a0, userInput
    li $t9, 100
    loopReset2:
        beqz $t9, end_loopReset
        addi $t9, $t9, -1
        sb $zero, ($a0)
        addi $a0, $a0, 1
        j loopReset2
    end_loopReset:
    li $v0, 10
    syscall
#-------------------------------------------------------------------------------------------
# reset lai phan bo nho khong phuc vu luu tru palindromes va quay tro lai ham main
#-------------------------------------------------------------------------------------------

goAgain:
    la $a0, userInput
    li $t9, 100
    loopReset:
        beqz $t9, main
        addi $t9, $t9, -1
        sb $zero, ($a0)
        addi $a0, $a0, 1
        j loopReset
```

## Kết quả chạy:


Mars Messages
Run I/O
Clear
Enter a string to check if it is a palindrome: trung
trung is not a palindrome :(
Would you like to enter another? (yes/no): y
Enter a string to check if it is a palindrome: ahalaha
ahalaha is a palindrome :)
Would you like to enter another? (yes/no): y
Enter a string to check if it is a palindrome: ahalaha
The String just have entered is exist in Palindromes
Would you like to enter another? (yes/no): y
Enter a string to check if it is a palindrome: aaaaahhhhhhhhhhhhhhhhfhfhffhfhfhfhfhfhfhfhfhfh
Length of string is too long
Would you like to enter another? (yes/no): y
Enter a string to check if it is a palindrome: huhfe*hde
The string you just entered contains a special character
Would you like to enter another? (yes/no): n
Program is finished.
-- program is finished running --

# Project 2:

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

Input: N = 3, M = 11

Output: 3, 5, 7, 11

**Phân tích đề bài:**

- Input là 2 số nguyên không âm
- Output là các số nguyên tố nằm ở trong khoảng đã nhập

**Phân tích cách thực hiện:**

- Ý tưởng thực hiện:

- Nhập vào số N và M
- Thực hiện thuật toán tìm kiếm số nguyên tố trong khoảng [N;M]
- In ra màn hình các số nguyên tố tìm được

- Các bước thực hiện:

- Bước 1: Nhập N:

Nếu N không phải số nguyên dương thì nhập lại.

- Bước 2: Nhập M:

Nếu M không phải số nguyên dương thì nhập lại.

- Bước 3: Thực hiện thuật toán tìm kiếm số nguyên tố

Thuật to:

```
input = enter from keyboard

for( i = N; i <= M; i++){
  if(i==1 || i==0) continue;
  flag = 1;
  for( j = 2; j < i/2; j++)
    if(i%j == 0) {flag = 0; break;} //Tìm ước của các số
  print(resual)
}
```

- Bước 4: In ra màn hình các số nguyên tố tìm được

**Ý nghĩa của các thanh ghi được sử dụng:**

- $s0: Lưu giá trị biến N
- $s1: Lưu giá trị biến M
- $s2: Lưu giá trị biến i

- $s3: Lưu giá trị biến j
- $s4: Lưu giá trị biến flag
- $s5: Lấy phần nguyên khi chia
- $a0: Nhập vào xuất các giá trị
- $t0, $t1, $t2: Biến tạm dùng khi so sánh

**Ý nghĩa của các thanh ghi được sử dụng:**

- check_var: kiểm tra kiểu của số nhập vào N và M
- check_prime_number: duyệt từ số N đến M tìm và in số nguyên tố

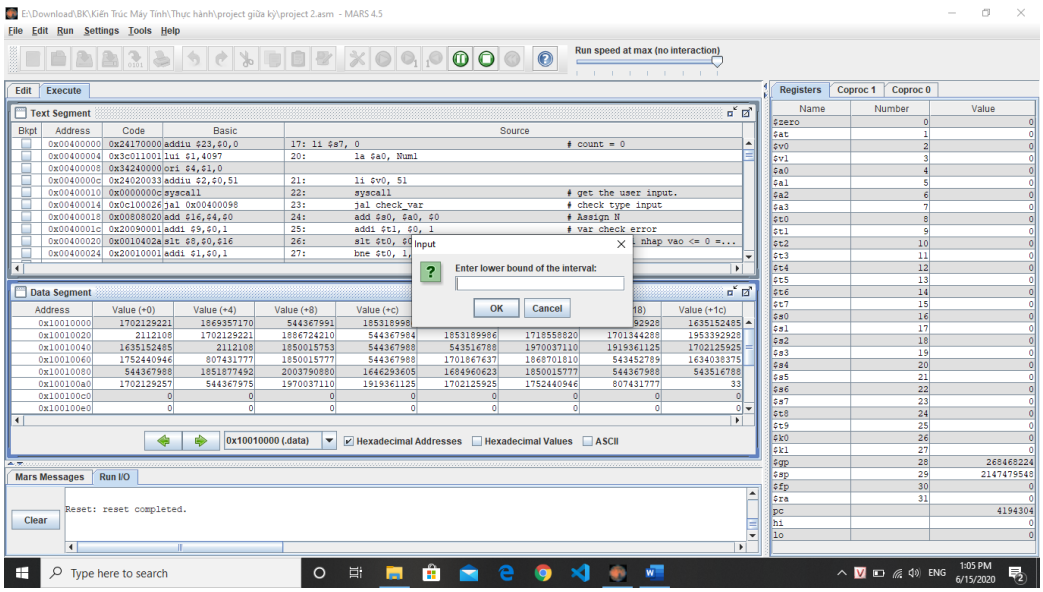
**Mã nguồn:**

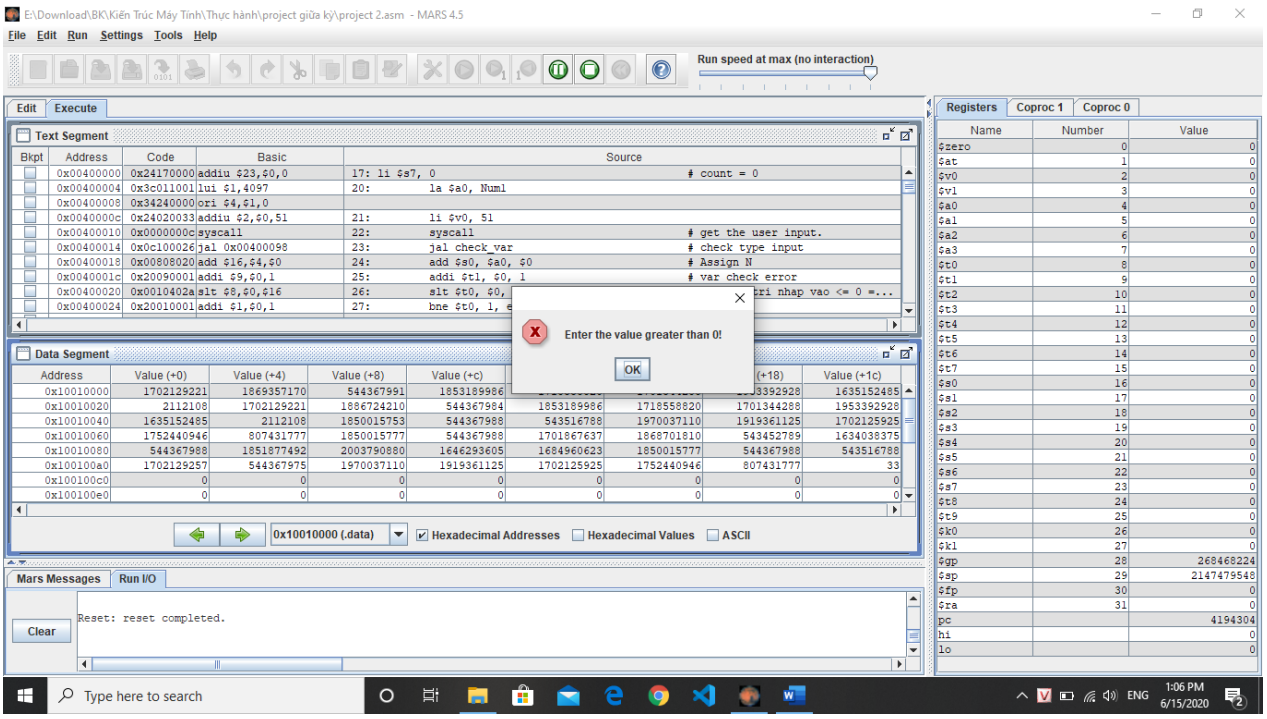
```
# $s0 = N, $s1 = M, $s2 = i, $s3 = j, $s4 = flag
.data
Num1: .asciiz "Enter lower bound of the interval: "
Num2: .asciiz "Enter upper bound of the interval: "
a: .asciiz "\t"
error1: .asciiz "Enter the value greater than 0!"
error2: .asciiz "Enter upper bound greater than lower bound!"
error3: .asciiz "Enter the integer value greater than 0!"

.text
#-----------------------------------------------------------------------------
# input1, input2: Nhap khoang kiem tra
# error_input_1, error_input_2, error_input_3: Kiem tra loi dau vao
# error1: Check input type
#-----------------------------------------------------------------------------
main:
li $s7, 0                               # count = 0

input1:
      la $a0, Num1
      li $v0, 51
      syscall                           # get the user input.
      jal check_var                     # check type input
      add $s0, $a0, $0                  # Assign N
      addi $t1, $0, 1                         # var check error
      slt $t0, $0, $s0                  # neu gia tri nhap vao <= 0 => jump input1
      bne $t0, 1, error_input

input2:      la $a0, Num2
      li $v0, 51
      syscall                           # get the user input.
      jal check_var                     # check type input
      add $s1, $a0, $0                  # Assign M
      slt $t0, $0, $s1                  # neu gia tri nhap vao < 0 => jump input2
      bne $t0, 1, error_input
```

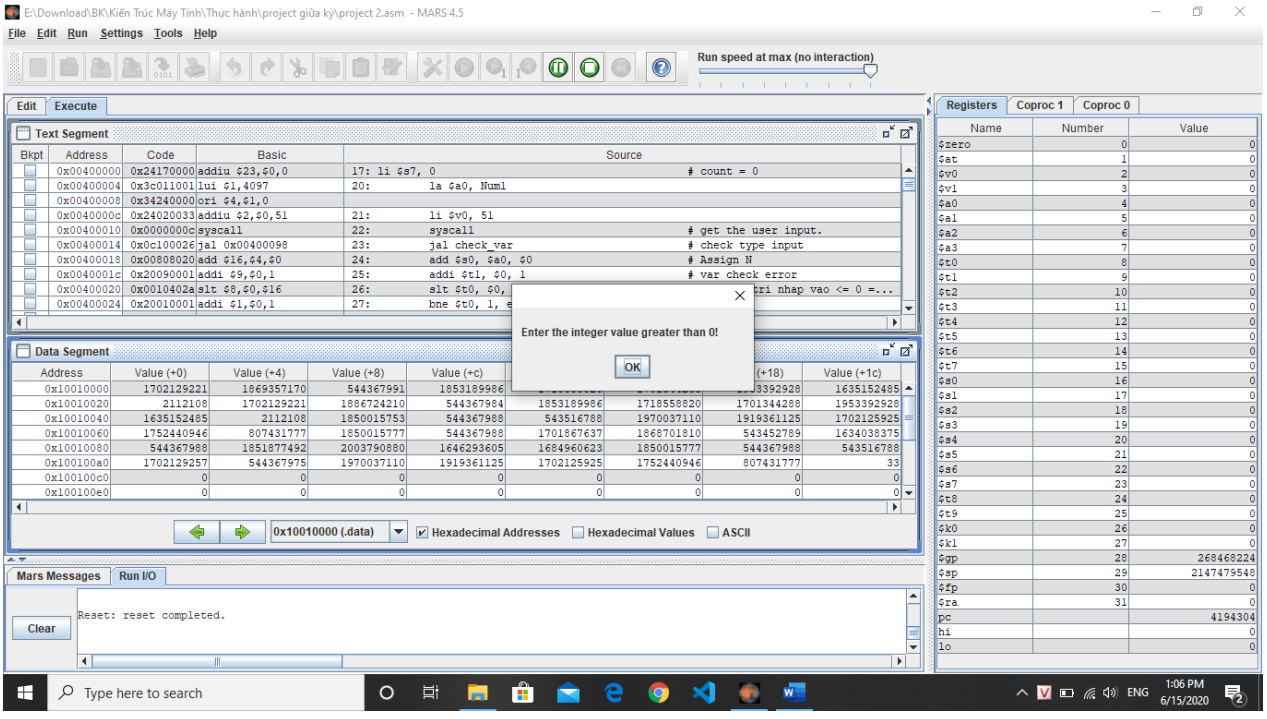
```
	slt $t0, $s1, $s0		# Check upper greater than lower???
	bne $t0, $0, error_input_2
	add $s2, $0, $s0		# i = N

	jal check_prime_number		# Nhay den ham kiem tra so nguyen to
	li $v0, 10
	syscall				# Ket thuc chuong trinh
error_input:
	la $v0, 55
	la $a0, error1
	syscall				# In thong bao loi input <0
	beq $t1, 1, input2
	j input1
error_input_2:
	la $v0, 55
	la $a0, error2
	syscall				# In thong bao loi upper be hon lower
	j input2

check_var:
	beq $a1, -1, error_1		#
	beq $a1, -2, error_1		#
	beq $a1, -3, error_1		#
	jr $ra				# Check loi input
	error_1:
		la $v0, 55
		la $a0, error3
		syscall			# In thong bao loi kieu nhap vao
		beq $t2, 1, input2
		j input1
#---------------------------------------------------------------------------
# for_1, for_2: Dung vong lap duyet tung so tu N->M de tim so nguyen to
# for( i = N; i <= M; i++){
#	if(i==1 || i==0) continue;
#	flag = 1;
#	for( j = 2; j < i/2; j++)
#		if(i%j == 0) {flag = 0; break;} //Tìm ??c c?a các s?
#	if(flag ==1) print(resual);
# }
#---------------------------------------------------------------------------

check_prime_number:
for_1:
	slt $t0, $s1, $s2		#
	bne $t0, $0, end_for_1		# Neu i > M ket thuc vong lap 1
	beq $s2, 1, end_for_2		# Neu i = 1 thi tang i
	addi $s3, $0, 2			# j = 2
	addi $t9, $0, 2			# Bien tam k = 2
	div $s2, $t9			#
	mflo $s5			# $s5 = i div 2
	addi $s4, $0, 1			# set flag = 1
```

```
	for_2:
		slt $t1, $s5, $s3		#
		bne $t1, 0, end_for_2		# Neu j > i/2 ket thuc vong lap 2, tang i
		div $s2, $s3			#
		mfhi $t2			# $t2 = i mod 3
		beq $t2, $0, push		# Neu $t2 = 0 -> không phai so nguyen to
		addi $s3, $s3,1			# j++
		j for_2
	print:
		beq $s7, 51, end_for_1		# Kiem tra bien count, count > 50 dung in,
thoat chuong trinh
		add $a0, $s2, $0		#
		li $v0, 1			#
		syscall				# In so nguyen to
		li $v0, 4			#
		la $a0, a			#
		syscall				# Dau tab
		addi $s7, $s7, 1		# count ++

		addi $s2, $s2, 1		# i++
		j for_1
	push:  # gan flag = 0
		add $s4, $0, $0			# flag = 0
		addi $s2, $s2, 1		# i++
		j for_1
	end_for_2: #quay lai vong lap 1
		beq $s4, 1, print		# Neu flag = 1, in so nguyen to
		addi $s2, $s2, 1		# i++
		j for_1
end_for_1: #ket thuc chuong trinh
	jr $ra
```

## Kết quả chạy:






```
11      13      17      19      23      29      31      37      41      43      47      53      59      61      67      71      73
-- program is finished running --
```