Nguyễn Kiều Trang – 20205174 <6A – 8B – 6C>

6A.

```
.data
input: .asciiz "Input N: "
Error:    .asciiz "Interger should be positive! Please enter try again.\n"
result: .asciiz "The integer in octal system is: "
.align 0
res: .space 80
.text
main:
        #Input N
        la      $a0, input              #address of input integer
        li      $v0, 4                  #system call for string display
        syscall
        li      $v0, 5                  #read interger system call
        syscall
        move    $s0, $v0                #store first integer in s0

        #check positive
        blez    $s0, error
        #print result messenger
        la      $a0, result             #address of result mess
        li      $v0, 4
        syscall
        # Change
        li      $t0, 8
        la      $s1, res            # $s1 luu dia chi co so cua res
```

```
        li      $t2, 0          # bien dem index i
        Solv:
                div     $s0,$t0
                mflo    $s0     # thuong cap nhat vao $s0
                mfhi    $a0     # phan du = $t1
                pushStack:
                        sb      $a0,0($s1)
                        beqz    $s0,Exit
                        addi    $t2,$t2,1
                        add     $s1,$s1,1


                jal     Solv
error:
        la      $a0, Error      #address of error mess
        li      $v0, 4
        syscall
        j       main
Exit:
        li      $v0,1
        Loop:
                bltz    $t2,ExitLoop
                lb      $a0,0($s1)
                syscall
                addi    $t2,$t2,-1
                add     $s1,$s1,-1
                jal     Loop
        ExitLoop:
```

N = -2 -> Yêu cầu người dùng nhập lại số dương.

N = 100

```
Input N: -2
Interger should be positive! Please enter try again.
Input N: 100
The integer in octal system is: 144
-- program is finished running (dropped off bottom) --
```

N = 6

```
Input N: 6
The integer in octal system is: 6
-- program is finished running (dropped off bottom) --
```

N = 12

```
Input N: 12
The integer in octal system is: 14
-- program is finished running (dropped off bottom) --
```

B8

```
.data
inputN: .asciiz "Nhap so phan tu trong mang: "
inputArr: .asciiz "Nhap mang:\n"
error: .asciiz "So phan tu phai la so duong. Vui long nhạp lai !!!!\n"
mess1: .asciiz "So nguyen am lon nhat la: "
endl: .asciiz "\n"
mess2: .asciiz "Vi tri cua so do <tinh tu vi tri so 0>: "
.align 2
A: .word #mangsonguyen
.text

        la      $s0,A
        add     $t0,$zero,$s0
        li      $s2,-100000     # max hien tai
        li      $s3,0           # vi tri tim thay max
get_num:
        #so luong phan tu
        la      $a0, inputN             # In dong inputN
        li      $v0, 4
        syscall
        li      $v0, 5
        syscall

        blt     $v0,0,error1
        add     $s1,$zero,$v0   #s1 = n - so luong phan tu cua mang
        li      $t1,0           #index i =0
```

```
        la      $t0,A


# Nhap mang A
        la      $a0, inputArr           #In dong inputArr
        li      $v0, 4                  #system call for string display
        syscall
get_arr:
        beq     $t1,$s1,end_get_arr


        li      $v0, 5
        syscall
        bgez    $v0, continue
                check_max:
                        bgt     $v0,$s2,update_max
                        jal     continue
                        update_max:
                                move    $s2,$v0 # gan gtri v0 vao s2
                                move    $s3,$t1 # gan gtri t1 vao s3
        continue:
        sw      $v0,0($t0)
        addi    $t0,$t0,4       #
        addi    $t1,$t1,1       #i++
        j       get_arr


#In loi neu so phan tu trong mang be hon 0
error1:
        li      $v0, 4
        la      $a0, error
        syscall
```

```
        j       get_num
end_get_arr:
# in mess1
li      $v0,4
la      $a0,mess1
syscall
#in so am lon nhat
li      $v0, 1
move    $a0,$s2
syscall
# in \n
li      $v0,4
la      $a0,endl
syscall
# in mess2
li      $v0,4
la      $a0,mess2
syscall
# in vi tri index
li      $v0, 1
move    $a0,$s3
syscall
```

- Số phần tử trong mảng: -5
  Yêu cầu người dùng nhập lại.
  Số phần tử trong mảng là 6
  Arr = { -1, 0, 2, -8, -6, 3}
  ⇨ Kết quả: Số âm lớn nhất là: -1. Ở vị trí số 0 <tính từ vị trí 0 đến N – 1>.

```
Nhap so phan tu trong mang: -5
So phan tu phai la so duong. Vui long nh¡p lai !!!!
Nhap so phan tu trong mang: 6
Nhap mang:
-1
0
2
-8
-6
3
So nguyen am lon nhat la: -1
Vi tri cua so do <tinh tu vi tri so 0>: 0
-- program is finished running (dropped off bottom) --
```

- Tương tự

```
Nhap so phan tu trong mang: -1
So phan tu phai la so duong. Vui long nh¡p lai !!!!
Nhap so phan tu trong mang: 5
Nhap mang:
6
8
-5
-9
3
So nguyen am lon nhat la: -5
Vi tri cua so do <tinh tu vi tri so 0>: 2
-- program is finished running (dropped off bottom) --
```

C6

```
.data
mss1: .asciiz "\nNhap xau: "
mss2: .asciiz "\nNhap ky tu C: "
result: .asciiz "\n\n=>>So phan tu C trong chuoi la: "
.align 0
string: .space 100
.text
#get_string
	la	$t1,string
get_sting:
	li	$v0,4
	la	$a0,mss1
	syscall
	li	$v0, 8
	la	$a0, string
	li	$a1, 100
	syscall
 get_char:
	li	$v0,4
	la	$a0,mss2
	syscall
	li	$v0, 12
	syscall
	move	$s0,$v0 # $s0 luu gia tri cua ky tu can tim
	#lay do dai xau da nhap
	li	$t0,0	# count
	la	$t1,string
get_length:
```

```
        lb      $t2,0($t1)
        beq     $t2, $zero, end_of_str    # is null char?
        addi    $t0, $t0, 1               # $t0 = $t0 + 1 -> i = i + 1
        addi    $t1,$t1,1
        j       get_length
#xu li do dai xau, de thuc hien vong lap
end_of_str:
        move    $s2,$t0  # $s2 luu so luong ky tu trong xau
count_sque:
        li      $t0,0           # index loop i
        la      $t1,string      # dia chi co so cua xau
        li      $s3,0
        loop:
                lb      $t2,0($t1)
                beq     $t2, $zero, end_of_loop # is null char?
                xor     $t3, $t2, 0x20          # ky tu in hoa
                addi    $t0, $t0, 1             # $t0 = $t0 + 1 -> i = i + 1
                addi    $t1,$t1,1
                beq     $s0,$t2,update_count
                beq     $s0,$t3,update_count
                jal     loop
                update_count:           addi    $s3,$s3,1
                                        jal     loop
        end_of_loop:
        li      $v0,4
        la      $a0,result
        syscall
        li      $v0,1
        move    $a0,$s3
        syscall
```

```
Nhap xau: NguyenKieuTranggGGGg

Nhap ky tu C: g

=>>So phan tu C trong chuoi la: 7
-- program is finished running (dropped off bottom) --
```

```
Nhap xau: 22Xinchao2

Nhap ky tu C: 2

=>>So phan tu C trong chuoi la: 3
-- program is finished running (dropped off bottom) --
```