

# Báo cáo giữa kỳ

## Bài 1:

# 5. Nhập số nguyên dương N,  
# in ra màn hình tổng các chữ số là số lẻ  
# và tổng các chữ số là số chẵn của N.

.data

Message1: .ascii "Tong chan: "

Message2: .ascii "\nTong le: "

.text

li \$v0, 5 #nhap so nguyen N

syscall

add \$s1, \$0, \$0 # Tong chan = 0

add \$s2, \$0, \$0 # Tong le = 0

add \$s0, \$0, \$v0 # dat \$s0 = N

addi \$t0, \$0, 10 # \$t0 = 10

addi \$t3, \$0, 2 # \$t3 = 2

while: beqz \$s0, print # \$s0 = 0 thì j check 3

nop

div \$s0, \$t0 # lay \$s0 chia cho 10

mflo \$s0 # chuyen phan thuong vao \$s0

mfhi \$t1 # chuyen phan du vao \$t1

div \$t1, \$t3 # \$t1/2

mfhi \$t4 # chuyen phan du vao \$t4

j check1

nop

check1:

beqz \$t4, TongChan # neu phan du \$t4 = 0 thi jumb tong chan

nop

add \$s2, \$s2, \$t1 # else cong \$t1 vao tong le

j while

nop

TongChan:

add \$s1, \$s1, \$t1

j while

nop

print:

# in ra dong Tong chan

li \$v0, 4

la \$a0, Message1

syscall

#in ra gia tri tong chan

move \$a0, \$s1

li \$v0, 1

syscall

#in ra dong Tong le

li \$v0, 4

la \$a0, Message2

syscall

#in ra gia tri tong le

move \$a0, \$s2

li \$v0, 1

syscall

done:

Kết quả:

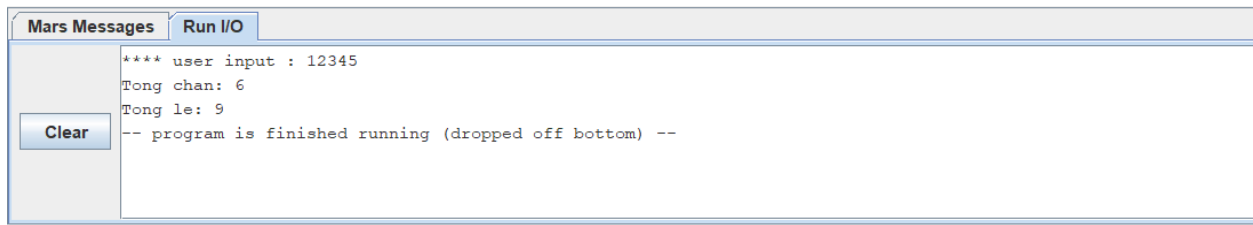
The screenshot displays the MIPS simulator interface. The 'Text Segment' window shows the following assembly code:

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24020005	addiu \$2,\$0,5	8: li \$v0, 5 #nhap so nguyen N
	0x00400004	0x0000000c	syscall	9: syscall
	0x00400008	0x00008820	add \$17,\$0,\$0	11: add \$s1, \$0, \$0 # Tong chan = 0
	0x0040000c	0x00009020	add \$18,\$0,\$0	12: add \$s2, \$0, \$0 # Tong le = 0
	0x00400010	0x00028020	add \$16,\$0,\$2	13: add \$a0, \$0, \$v0 # dat \$a0 = N
	0x00400014	0x2008000a	addi \$8,\$0,10	14: addi \$t0, \$0, 10 # \$t0 = 10
	0x00400018	0x200b0002	addi \$11,\$0,2	15: addi \$t3, \$0, 2 # \$t3 = 2
	0x0040001c	0x12000010	beg \$16,\$0,16	16: while: begz \$a0, print #\$a0 = 0 thi j check 3
	0x00400020	0x00000000	nop	17: nop
	0x00400024	0x0208001a	div \$16,\$8	18: div \$s0, \$t0 # lay \$s0 chia cho 10
	0x00400028	0x00008012	mflo \$16	19: mflo \$s0 # chuyen phan tuong vao \$s0
	0x0040002c	0x00004810	mfhi \$9	20: mfhi \$t1 # chuyen phan d

The 'Data Segment' window shows the following data:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)
0x10010000	1735290708	1634231072	2112110	1852789770
0x10010020	0	0	0	0
0x10010040	0	0	0	0
0x10010060	0	0	0	0
0x10010080	0	0	0	0
0x100100a0	0	0	0	0
0x100100c0	0	0	0	0
0x100100e0	0	0	0	0
0x10010100	0	0	0	0
0x10010120	0	0	0	0

A 'MIPS Keyboard Input' dialog box is open, prompting 'Enter an integer value (syscall 5)' with the input '12345' and an 'OK' button.



Ý tưởng là đầu tiên là em nhập vào số nguyên dương N, em dùng vòng lặp chia N cho 10 để lấy phần phần dư, và gán thương lấy được cho N. Tiếp tục lặp lại cho đến khi N bằng 0. Như vậy em sẽ có được các chữ số của N. Và sau đó thì dễ dàng tính được tổng chữ số chẵn, và tổng chữ số lẻ.

## Bài 2:

#2. Nhập mảng số nguyên từ bàn phím.

#Sắp xếp các phần tử có giá trị dương giảm dần.

.data

myArray: .word 0:1000

.text

li \$v0, 5 #nhap so nguyen N

syscall

la \$t0, myArray #lay dia chi co so cua mang

add \$t1, \$0, \$0 # i= 0

add \$t2, \$0, \$v0 # \$t2 = N

for: beq \$t1, \$t2, done # neu i = N jumb done

sll \$t3, \$t1, 2 # \$t3 = i \* 4

add \$t3, \$t3, \$t0 # dia chi cua myArray[i]

j NhapMang # jumb NhapMang

tiếp:

```
sw $t4, 0($t3)
addi $t1, $t1, 1 # tang i len 1
j for #nhay den for
```

NhapMang:

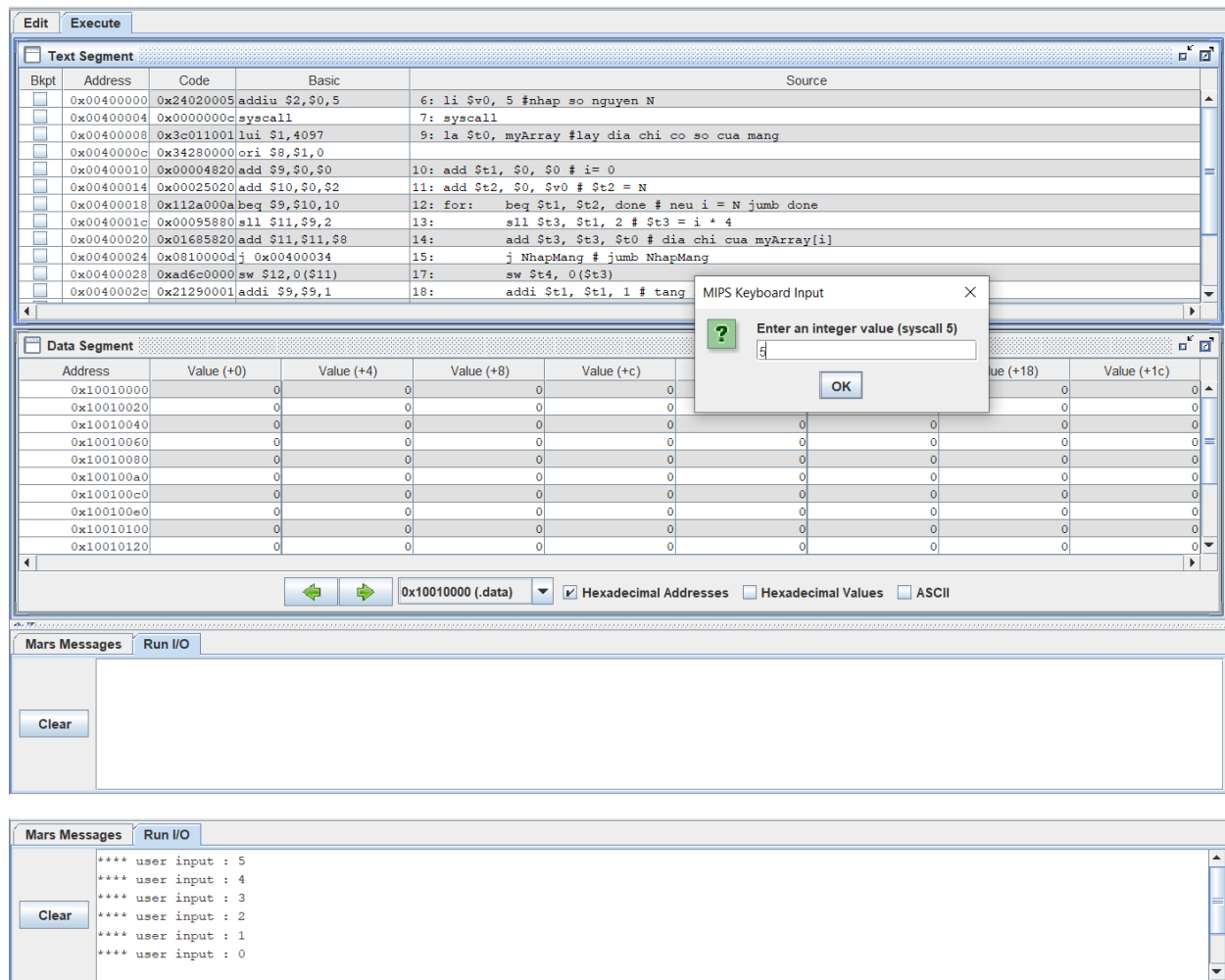
```
li $v0, 5 #nhap so nguyen N
syscall
move $t4, $v0
j tiep
```

#reset:

```
#    add $t1, $0, $0 # i = 0
#    add $t5, $0, $0 # j = 0
#
#for1: beq $t1, $t2, done # neu i = N thoat
#    sll $t3, $t1, 2 # $t3 = i * 4
#for2: beq $t5
#    add $t3, $t3, $t0 # dia chi cua myArray[i]
#    lw $t4, 0($t3)
#    addi $t1, $t1, 1 # tang i len 1
#    j for1 #nhay den for1
```

done:

Kết quả thực hiện:



Như trên hình cái box đầu tiên mình sẽ nhập số phần tử của mảng(N)  
và N box tiếp theo mình sẽ nhập từng phần tử cho mảng

### Bài 3:

# 6. Nhập vào chuỗi ký tự. In ra số từ trong chuỗi bắt đầu bằng 2 ký tự "Ch".

.data

A: .space 100 # Buffer 100 byte chứa chuỗi kí tự cần

.text

```
li $v0, 8
la $a0, A
li $a1, 100
syscall
```

```
add $s0,$zero,$zero # $s0 = i = 0
move $t5,$a0
add $t6, $0, $0 # $t6 = 0
add $s7, $0, $0 # $s7 = 0
```

L1:

```
add $t1,$s0,$t5 # $t1 = address of A[i]
lbu $t2,0($t1) # $t2 = value at $t1 = A[i]

addi $t7, $s0, 1 # $t7 = $s0 + 1
add $t7, $t7, $t5 # $t7 = address of A[i+1]
lbu $s6, 0($t7) # lay gia tri A[i+1]
beq $t2,$zero,print # if A[i] == 0, jump print
nop
j check1
nop
```

next:

```
nop
```

```
addi $s0,$s0,1 # $s0 = $s0 + 1 <-> i = i + 1
```

```
j L1 # next character
```

```
nop
```

```
check1:
```

```
beq $t2, 67, check2
```

```
nop
```

```
j next
```

```
nop
```

```
check2:
```

```
beq $s6, 104, count
```

```
nop
```

```
count:
```

```
addi $s7, $s7, 1 # tang $s7 them 1
```

```
move $a0, $s7
```

```
j next
```

```
nop
```

```
print:
```

```
li $v0, 1
```

```
syscall
```



done:

Kết quả thực hiện:

F:\20212\KienTrucMayTinh\Mars\BT\_ThucHanh\ThiGKy\mps4.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

**Text Segment**

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24020008	addiu \$2,\$0,8	5: li \$v0, 8
	0x00400004	0x3c011001	lui \$1,4097	6: la \$a0, A
	0x00400008	0x34240000	ori \$4,\$1,0	
	0x0040000c	0x24050064	addiu \$5,\$0,100	7: li \$a1, 100
	0x00400010	0x0000000c	syscall	8: syscall
	0x00400014	0x00008020	add \$16,\$0,\$0	10: add \$s0,\$zero,\$zero # \$s0 = i = 0
	0x00400018	0x00046821	addu \$13,\$0,\$4	11: move \$t5, \$a0
	0x0040001c	0x00007020	add \$14,\$0,\$0	12: add \$t6, \$0, \$0 # \$t6 = 0
	0x00400020	0x0000b820	add \$23,\$0,\$0	13: add \$t7, \$0, \$0 # \$t7 = 0
	0x00400024	0x020d4820	add \$9,\$16,\$13	15: add \$t1,\$s0,\$t5 # \$t1 = address of A[i]
	0x00400028	0x912a0000	lbu \$10,0(\$9)	16: lbu \$t2,0(\$t1) # \$t2 = value at \$t1 = A[i]
	0x0040002c	0x220f0001	addi \$15,\$16,1	18: addi \$t7, \$s0, 1 # \$t7 = i

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)
0x10010000	0	0	0	0
0x10010020	0	0	0	0
0x10010040	0	0	0	0
0x10010060	0	0	0	0
0x10010080	0	0	0	0
0x100100a0	0	0	0	0
0x100100c0	0	0	0	0
0x100100e0	0	0	0	0
0x10010100	0	0	0	0
0x10010120	0	0	0	0

MIPS Keyboard Input

Enter a string of maximum length 99 (syscall 8)

Cham chi

OK

Mars Messages Run I/O

Clear

```
**** user input : Cham chi
1
-- program is finished running (dropped off bottom) --
```

6. Nhập vào xâu ký tự. In ra số từ trong xâu bắt đầu bằng 2 ký tự “Ch”.

Ở đây có ký tự ‘C’ viết hoa nên nếu đúng chính tả thì sẽ không có trường hợp “Ch” nằm ở cuối 1 từ

vì vậy em dùng vòng lặp kiểm tra nếu có đồng thời  $A[i] = 'C'$  và  $A[i+1] = 'h'$

thì em sẽ tăng biến đếm lên 1 tức là có thêm 1 từ bắt đầu bằng Ch