

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**



**UIT**  
Trường Đại học  
Công nghệ Thông tin

**Khoa Khoa học  
và Kỹ thuật Thông tin**

**Nguyễn Lê Quỳnh Hương – 21520255**

IT012.N11.CNCL

# **BÁO CÁO: Lab 5**

**GVHD: Nguyễn Văn Tín**

**Hồ Chí Minh, 2022**

## 1. Thao tác với mảng

Viết code trong phần “.text” thực hiện riêng từng phần việc:

✓ In ra cửa sổ I/O của MARS tất cả các phần tử của mảng array1 và array2

<pre>.data  array1:.word 5,6,7,8,1,2,3,9,10,4  size1: .word 10   array2:.byte 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16  size2: .word 16   array3:.space 8  size3: .word 8   space:.asciiz " "  newLine:.asciiz "\n"  .text</pre>	<pre>#Khai báo vùng nhớ chứa data   #Khai báo vùng nhớ chứa mã lệnh</pre>
--	---

<pre> la \$a1, array1  lw \$a2, size1  li \$t0, 0  jal PrintArray1   li \$v0, 4  la \$a0, newLine  syscall   la \$a1, array2  lw \$a2, size2  li \$t0, 0  jal PrintArray2  j End  PrintArray1:      blt \$t0, \$a2, LoopPrintArray1      jr \$ra  LoopPrintArray1:      li \$v0, 1      lw \$a0, 0(\$a1)      syscall </pre>	<pre> #index = 0  #PrintArray1(array1, size1)   #syscall in dòng mới  #PrintArray2(array2, size2)   #if(index &lt; size1) =&gt; LoopPrintArr  #return </pre>
--	--

<pre> li \$v0, 4  la \$a0, space  syscall  addi \$a1, \$a1, 4  addi \$t0, \$t0, 1  j PrintArray1  PrintArray2:      blt \$t0, \$a2, LoopPrintArray2      jr \$ra  LoopPrintArray2:      li \$v0, 1      lb \$a0, 0(\$a1)      syscall      li \$v0, 4      la \$a0, space      syscall      addi \$a1, \$a1, 1      addi \$t0, \$t0, 1      j PrintArray2  End: </pre>	<pre> #Chuyển phần tử tiếp theo  #if(index &lt; size2) =&gt; LoopPrintArr  #return  #Chuyển phần tử tiếp theo </pre>
--	--

✓ Gán các giá trị cho mảng array3 sao cho :  

$$\text{array3}[i] = \text{array2}[i] + \text{array2}[\text{size2} - 1 - i]$$

<pre>.data  array1:.word 5,6,7,8,1,2,3,9,10,4  size1: .word 10   array2:.byte 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16  size2: .word 16   array3:.space 8  size3: .word 8   .text  li \$t0, 0  lw \$s2, size2  lw \$s4, size3  j IfLoop  IfLoop:  blt \$t0, \$s4, Loop  j End</pre>	<pre>#Khai báo vùng nhớ chứa data   #Khai báo vùng nhớ chứa mã lệnh   #if(index &lt; size3) ==&gt; Loop</pre>
---	---

<p>Loop:</p> <pre> la \$s1, array2  la \$s3, array3  add \$s0, \$s1, \$t0  lb \$t9, 0(\$s0)   subi \$t8, \$s2, 1  sub \$t8, \$t8, \$t0  add \$s0, \$s1, \$t8  lb \$t8, 0(\$s0)   add \$t9, \$t8, \$t9  sll \$t1, \$t0, 2  add \$s3, \$s3, \$t1  sw \$t9, 0(\$s3)   addi \$t0, \$t0, 1  j IfLoop  End: </pre>	<pre> #array2[i]   #array2[size2-1-i]   #array2[i] + array2[size2 - 1 - i]   #Chuyển phần tử tiếp theo #array3[i] = ...   #index++ </pre>
--	---

✓ Người sử dụng nhập vào mảng thứ mấy và chỉ số phần tử cần lấy trong mảng đó, chương trình xuất ra phần tử tương ứng.

<pre> .data  array1:.word 5,6,7,8,1,2,3,9,10,4  size1: .word 10   array2:.byte 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16  size2: .word 16   array3:.space 8  size3: .word 8   inputArray: .ascii "Nhap mang can tim (chi nha so): "  inputPos:.ascii "Nhap vao chi so: "  error:      .ascii "Khong ton tai chi so nay"  .text  li \$v0, 4  la \$a0, inputArray </pre>	<pre> #Khai báo vùng nhớ chứa data   #Khai báo vùng nhớ chứa mã lệnh </pre>
---	---

<pre> syscall  li \$v0, 5  syscall  move \$t0, \$v0   li \$v0, 4  la \$a0, inputPos  syscall  li \$v0, 5  syscall  move \$t1, \$v0   beq \$t0, 1, Array1  beq \$t0, 2, Array2  Array1:  la \$s0, array1  lw \$s1, size1  bge \$t1, \$s1, Error  sll \$t1, \$t1, 2  add \$s0, \$s0, \$t1  lw \$a0, 0(\$s0) </pre>	<pre> #In array1   #if(I &gt; size) =&gt; Error   #arr1[i] = ... </pre>
--	---



<pre> li \$v0, 1  syscall  j End Array2:  la \$s0, array2  lw \$s1, size2  bge \$t1, \$s1, Error  add \$s0, \$s0, \$t1  lb \$a0, 0(\$s0)  li \$v0, 1  syscall  j End Error:  li \$v0, 4  la \$a0, error  syscall  j End End: </pre>	<pre> #In array2  #if(I &gt; size) =&gt; Error  #arr2[i] = ...  #In lỗi </pre>
---	--

## 2. Thao tác với con trỏ

<pre>.data  array1:.word 5,6,7,8,1,2,3,9,10,4  size1: .word 10   array2:.byte 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16  size2: .word 16   array3:.space 8  size3: .word 8   space: .ascii " "  newLine:.ascii "\n"   .text  la \$a1, array1  lw \$a2, size1</pre>	<pre>#Khai báo vùng nhớ chứa data   #Khai báo vùng nhớ chứa mã lệnh   #index = 0</pre>
---	--

<pre> li \$t0, 0  jal PrintArray1   li \$v0, 4  la \$a0, newLine  syscall   la \$a1, array2  lw \$a2, size2  li \$t0, 0  jal PrintArray2  j End  PrintArray1:      blt \$t0, \$a2, LoopPrintArray1      jr \$ra  LoopPrintArray1:      li \$v0, 1      lw \$a0, 0(\$a1)      syscall      li \$v0, 4      la \$a0, space </pre>	<pre> #PrintArray1(array1, size1)   #syscall in dòng mới   #PrintArray2(array2, size2)   #if(index &lt; size1) =&gt; LoopPrintArr  #return </pre>
---	---

<pre> syscall  addi \$a1, \$a1, 4  addi \$t0, \$t0, 1  j PrintArray1  PrintArray2:      blt \$t0, \$a2, LoopPrintArray2      jr \$ra  LoopPrintArray2:      li \$v0, 1      lb \$a0, 0(\$a1)      syscall      li \$v0, 4      la \$a0, space      syscall      addi \$a1, \$a1, 1      addi \$t0, \$t0, 1      j PrintArray2  End: </pre>	<pre> #Chuyển phần tử tiếp theo  #if(index &lt; size2) =&gt; LoopPrintArr  #return  #Chuyển phần tử tiếp theo </pre>
--	--

<pre> .data  array1:.word 5,6,7,8,1,2,3,9,10,4  size1: .word 10   array2:.byte 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16  size2: .word 16   array3:.space 8  size3: .word 8   .text  li \$t0, 0  lw \$s2, size2  lw \$s4, size3  j IfLoop  IfLoop:  blt \$t0, \$s4, Loop  j End  Loop:  la \$s1, array2 </pre>	<pre> #Khai báo vùng nhớ chứa data   #Khai báo vùng nhớ chứa mã lệnh   #if(index &lt; size3) =&gt; Loop </pre>
---	--

<pre> la \$s3, array3  add \$s0, \$s1, \$t0  lb \$t9, 0(\$s0)   subi \$t8, \$s2, 1  sub \$t8, \$t8, \$t0  add \$s0, \$s1, \$t8  lb \$t8, 0(\$s0)   add \$t9, \$t8, \$t9  sll \$t1, \$t0, 2  add \$s3, \$s3, \$t1  sw \$t9, 0(\$s3)   addi \$t0, \$t0, 1  j IfLoop  End: </pre>	<pre> #array2[i]   #array2[size2-1-i]   #array2[i] + array2[size2 - 1 - i]   #Chuyển phần tử tiếp theo #array3[i] = ...   #index++ </pre>
--	---

<pre> .data  array1:.word 5,6,7,8,1,2,3,9,10,4  size1: .word 10   array2:.byte 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16  size2: .word 16   array3:.space 8  size3: .word 8   inputArray: .ascii "Nhap mang can tim (chi nha so): "  inputPos:.ascii "Nhap vao chi so: "  error:      .ascii "Khong ton tai chi so nay"   .text  li \$v0, 4  la \$a0, inputArray  syscall  li \$v0, 5 </pre>	<pre> #Khai báo vùng nhớ chứa data   #Khai báo vùng nhớ chứa mã lệnh </pre>
---	---

<pre> syscall  move \$t0, \$v0   li \$v0, 4  la \$a0, inputPos  syscall  li \$v0, 5  syscall  move \$t1, \$v0   beq \$t0, 1, Array1 beq \$t0, 2, Array2 Array1: la \$s0, array1 lw \$s1, size1 bge \$t1, \$s1, Error sll \$t1, \$t1, 2 add \$s0, \$s0, \$t1 lw \$a0, 0(\$s0) li \$v0, 1 syscall </pre>	<pre> #In array 1   #if(I &gt; size) =&gt; Error   #arr1[i] = ... </pre>
--	--



<pre> j End  Array2:      la \$s0, array2      lw \$s1, size2      bge \$t1, \$s1, Error      add \$s0, \$s0, \$t1      lb \$a0, 0(\$s0)      li \$v0, 1      syscall  j End  Error:      li \$v0, 4      la \$a0, error      syscall  j End  End: </pre>	<pre> #In array2  #if(I &gt; size) =&gt; Error  #arr2[i] = ...  #In lỗi </pre>
---	--

### 3. Bài tập

a. Nhập một mảng các số nguyên n phần tử (nhập vào số phần tử và giá trị của từng phần tử), xuất ra cửa sổ I/O của MARS theo từng yêu cầu sau:

✓ Xuất ra giá trị lớn nhất và nhỏ nhất của mảng

✓ Tổng tất cả các phần tử của mảng

✓ Người sử dụng nhập vào chỉ số của một phần tử nào đó và giá trị của phần tử đó được in ra cửa sổ

<code>.data</code>  <code>Array:.space 100</code>  <code>InputN: .asciiz "Nhap N: "</code>  <code>MaxArr: .asciiz "\nMax = "</code>  <code>MinArr: .asciiz "\nMin = "</code>  <code>SumArr: .asciiz "\nSum = "</code>  <code>inputPos:.asciiz "\nNhap vao chi so: "</code>	<code>#Khai báo vùng nhớ chứa data</code>
<code>.text</code>  <code>la \$a0, InputN</code>  <code>li \$v0, 4</code>  <code>syscall</code>  <code>li \$v0, 5</code>  <code>syscall</code>  <code>move \$a2, \$v0</code>	<code>#Khai báo vùng nhớ chứa mã lệnh</code>    <code>#syscall nhập</code>    <code>#\$a2 = N</code>

<pre> la \$a1, Array li \$t0, 0 jal InputArray  la \$a1, Array li \$t0, 0 lw \$t1, 0(\$a1) jal MaxArray li \$v0, 4 la \$a0, MaxArr syscall li \$v0, 1 move \$a0, \$t1 syscall  la \$a1, Array li \$t0, 0 lw \$t1, 0(\$a1) jal MinArray li \$v0, 4 </pre>	<pre> #InputArray(Array, N)  #\$t1 = Max #MaxArray(Array, N, Max)  #Print Max  #\$t1 = Min #MinArray(Array, N, Min) </pre>
--	--

<pre> la \$a0, MinArr  syscall  li \$v0, 1  move \$a0, \$t1  syscall   la \$v0, 4  la \$a0, inputPos  syscall  li \$v0, 5  syscall  la \$a1, Array  sll \$v0, \$v0, 2  add \$a1, \$a1, \$v0  lw \$a0, 0(\$a1)  li \$v0, 1  syscall   j End  InputArray:  blt \$t0, \$a2, LoopInput </pre>	<pre> #Print Min Value   #Input vị trí mảng cần xuất   #Nhảy đến vị trí cần xuất   #Xuất giá trị </pre>
---	---

<pre> jr \$ra  LoopInput:      li \$v0, 5      syscall       sw \$v0, (\$a1)      add \$a1, \$a1, 4      addi \$t0, \$t0, 1      j InputArray  MaxArray:      blt \$t0, \$a2, FindMax      jr \$ra  FindMax:      add \$a1, \$a1, 4      lw \$t2, (\$a1)      bgt \$t2, \$t1, GanMax      addi \$t0, \$t0, 1      j MaxArray  GanMax:      move \$t1, \$t2      addi \$t0, \$t0, 1 </pre>	<pre> #Nhảy đến vị trí tiếp theo  #a[i+1] &gt; a[i] =&gt; GanMax  #Swap(A[i], A[i+1]) </pre>
---	--

<pre> j MaxArray  MinArray:  blt \$t0, \$a2, FindMin  jr \$ra  FindMin:      lw \$t2, (\$a1)      add \$a1, \$a1, 4      blt \$t2, \$t1, GanMin      addi \$t0, \$t0, 1      j MinArray  GanMin:      move \$t1, \$t2      addi \$t0, \$t0, 1      j MinArray  End: </pre>	<pre> # a[i+1] &lt; a[i] =&gt; GanMin  # Swap(A[i], A[i+1]) </pre>
--	--

b. Nhập một mảng các số nguyên n phần tử (nhập vào số phần tử và giá trị của từng phần tử). Mảng này gọi là A.

Chuyển dòng lệnh C dưới đây sang mã assembly của MIPS. Với các biến nguyên i, j được gán lần lượt vào thanh ghi \$s0, \$s1; và địa chỉ nền của mảng số nguyên A được lưu trong thanh ghi \$s3

if (i<j) A[i]= i;

else A[i] = j;

if (i<j) A[i]= i;	blt \$s0,\$s1,L    #so sanh \$s0<\$s1 đung thi nhay sang L
else A[i] = j;	sw \$s1,(\$s3)#A[i]=\$s1
	j exit
	L:
	sw \$s0,(\$s3)#A[i]=\$s0
	exit: