|  |  |
| --- | --- |
|  | **MIPS Assembly – Lab Solutions** |

**Exercise 1**

**Step 1**

.text 0x00400000

li $v0,4

la $a0,msg

syscall

li $v0,10

syscall

.data

msg: .asciiz "My first program"

**Exercise 2**

|  |  |
| --- | --- |
| **Step 1**  .text 0x00400000  #Section 1  li $v0,4  la $a0,msg1  syscall  #Section 2  li $v0,5  syscall  move $t0,$v0  #Section 3  li $v0,4  la $a0,msg2  syscall  #Section 4  li $v0,1  move $a0,$t0  syscall  #Section 5  li $v0,10  syscall  .data  msg1: .asciiz "Enter number:"  msg2: .asciiz "The number is:"  **Step 7**  .text 0x00400000  #read numbers  li $v0,5  syscall  move $t1,$v0  li $v0,5  syscall  move $t2,$v0  li $v0,5  syscall  move $t3,$v0  #display numbers  li $v0,4  la $a0,msg2  syscall  li $v0,1  move $a0,$t1  syscall  li $v0,4  la $a0,msg2  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,msg2  syscall  li $v0,1  move $a0,$t3  syscall  li $v0,10  syscall  .data  msg2: .asciiz "\nNumber="  **Step 9**  .text 0x00400000  # read numbers  li $v0,5  syscall  move $t1,$v0  li $v0,5  syscall  move $t2,$v0  add $t0,$t1,$t2  #display sum  li $v0,4  la $a0,msg  syscall  li $v0,1  move $a0,$t0  syscall  li $v0,10  syscall  .data  msg: .asciiz "(X+Y)=" | **Step 4**  .text 0x00400000  #read numbers  li $v0,4  la $a0,msg1  syscall  li $v0,5  syscall  move $t1,$v0  li $v0,4  la $a0,msg1  syscall  li $v0,5  syscall  move $t2,$v0  li $v0,4  la $a0,msg1  syscall  li $v0,5  syscall  move $t3,$v0  #display numbers  li $v0,4  la $a0,msg2  syscall  li $v0,1  move $a0,$t1  syscall  li $v0,4  la $a0,msg2  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,msg2  syscall  li $v0,1  move $a0,$t3  syscall  li $v0,10  syscall  .data  msg1: .asciiz "Enter number:"  msg2: .asciiz "\nNumber=" |

**Exercise 3**

|  |  |
| --- | --- |
| **Step 1,2**  .text 0x00400000  # read numbers  li $v0,5  syscall  move $t1,$v0  li $v0,5  syscall  move $t2,$v0  li $v0,5  syscall  move $t3,$v0  sub $t0,$t1,$t2  sub $t0,$t0,$t3  #display result  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t1  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,1  move $a0,$t3  syscall  li $v0,4  la $a0,mes3  syscall  li $v0,1  move $a0,$t0  syscall  li $v0,10  syscall  .data  mes1: .asciiz "("  mes2: .asciiz "-"  mes3: .asciiz ")="  **Step 9**  .text 0x00400000  # read numbers  li $v0,4  la $a0,ar1  syscall  li $v0,5  syscall  move $t1,$v0  li $v0,4  la $a0,ar2  syscall  li $v0,5  syscall  move $t2,$v0  li $v0,4  la $a0,ar3  syscall  li $v0,5  syscall  move $t3,$v0  #Z1  div $t0,$t1,$t2  add $t0,$t0,$t3  #Z2  div $t4,$t3,$t2  mul $t4,$t4,2  add $t4,$t4,$t1  #Z3  mul $t5,$t1,$t2  mul $t5,$t5,$t3  #Z4  sub $t6,$t2,$t1  div $t6,$t3,$t6  li $v0,4  la $a0,line  syscall  ####################Z1 parameterized  li $v0,4  la $a0,par1  syscall  li $v0,1  move $a0,$t1  syscall  li $v0,4  la $a0,dia  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,par2  syscall  li $v0,4  la $a0,syn  syscall  li $v0,1  move $a0,$t3  syscall  li $v0,4  la $a0,ison  syscall  li $v0,1  move $a0,$t0  syscall  ###################Z2 parameterized  li $v0,4  la $a0,line  syscall  li $v0,1  move $a0,$t1  syscall  li $v0,4  la $a0,syn  syscall  li $v0,1  li $a0,2  syscall  li $v0,4  la $a0,epi  syscall  li $v0,4  la $a0,par1  syscall  li $v0,1  move $a0,$t3  syscall  li $v0,4  la $a0,dia  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,par2  syscall  li $v0,4  la $a0,ison  syscall  li $v0,1  move $a0,$t4  syscall  ###################Z3 parameterized  li $v0,4  la $a0,line  syscall  li $v0,1  move $a0,$t1  syscall  li $v0,4  la $a0,epi  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,epi  syscall  li $v0,1  move $a0,$t3  syscall  li $v0,4  la $a0,ison  syscall  li $v0,1  move $a0,$t5  syscall  ####################Z4 parameterized  li $v0,4  la $a0,line  syscall  li $v0,1  move $a0,$t3  syscall  li $v0,4  la $a0,dia  syscall  li $v0,4  la $a0,par1  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,plin  syscall  li $v0,1  move $a0,$t1  syscall  li $v0,4  la $a0,par2  syscall  li $v0,4  la $a0,ison  syscall  li $v0,1  move $a0,$t6  syscall  li $v0,10  syscall  .data  ar1: .asciiz "Number1="  ar2: .asciiz " Number2 ="  ar3: .asciiz " Number3="  mes1: .asciiz "\nZ1="  mes2: .asciiz "\nZ2="  mes3: .asciiz "\nZ3="  mes4: .asciiz "\nZ4="  par1: .asciiz "("  par2: .asciiz ")"  dia: .asciiz "/"  syn: .asciiz "+"  ison: .asciiz "="  plin: .asciiz "-"  epi: .asciiz "\*"  line: .asciiz "\n" | **Step 7**  .text 0x00400000  # read numbers  li $v0,4  la $a0,ar1  syscall  li $v0,5  syscall  move $t1,$v0  li $v0,4  la $a0,ar2  syscall  li $v0,5  syscall  move $t2,$v0  li $v0,4  la $a0,ar3  syscall  li $v0,5  syscall  move $t3,$v0  #Z1  div $t0,$t1,$t2  add $t0,$t0,$t3  #Z2  div $t4,$t3,$t2  mul $t4,$t4,2  add $t4,$t4,$t1  #Z3  mul $t5,$t1,$t2  mul $t5,$t5,$t3  #Z4  sub $t6,$t2,$t1  div $t6,$t3,$t6  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t0  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,1  move $a0,$t4  syscall  li $v0,4  la $a0,mes3  syscall  li $v0,1  move $a0,$t5  syscall  li $v0,4  la $a0,mes4  syscall  li $v0,1  move $a0,$t6  syscall  li $v0,10  syscall  .data  ar1: .asciiz "Number1="  ar2: .asciiz "Number2="  ar3: .asciiz "Number3="  mes1: .asciiz "\nZ1="  mes2: .asciiz "\nZ2="  mes3: .asciiz "\nZ3="  mes4: .asciiz "\nZ4=" |

**Exercise 4**

|  |  |
| --- | --- |
| **Step 5a**  .text 0x00400000  li $v0,4  la $a0,mes  syscall  li $v0,5  syscall  move $t1,$v0  blt $t1,10,termination  bgt $t1,20,termination  li $v0,4  la $a0,msg  syscall  termination:  li $v0,10  syscall  .data  msg: .asciiz "$t1 belongs to [10,20]"  mes: .asciiz "give number:"  **Step 9**  .text 0x00400000  li $t1,1  start:  li $v0,4  la $a0,name  syscall  add $t1,$t1,1  ble $t1,15,start  li $v0,10  syscall  .data  name: .asciiz "\nPanayotis" | **Step 5b**  .text 0x00400000  li $v0,4  la $a0,mes  syscall  li $v0,5  syscall  move $t1,$v0  bge $t1,10,syn  j termination  syn:  ble $t1,20,syn2  j termination  syn2:  li $v0,4  la $a0,msg  syscall  termination:  li $v0,10  syscall  .data  msg: .asciiz "$t1 belongs to [10,20]"  mes: .asciiz "give number:"  **Step 10**  .text 0x00400000  li $t1,1  start:  ble $t1,15,loop  j termination  loop:  li $v0,4  la $a0,onoma  syscall  add $t1,$t1,1  j start  termination:  li $v0,10  syscall  .data  onoma: .asciiz "\nPanayotis" |

**Exercise 5**

|  |  |
| --- | --- |
| **Step 4**  .text 0x00400000  li $v0,4  la $a0,mes  syscall  li $v0,5  syscall  move $t1,$v0  li $v0,4  la $a0,mes  syscall  li $v0,5  syscall  move $t2,$v0  li $v0,4  la $a0,mes  syscall  li $v0,5  syscall  move $t3,$v0  add $t0,$t1,$t2  bgtz $t0,calculation1  beqz $t0,calculation2  #calculation3  mul $t4,$t3,$t3  sub $t4,$t2,$t4  add $t4,$t4,$t1  j termination  calculation1:  add $t4,$t3,3  mul $t4,$t4,$t1  j termination  calculation2:  mul $t4,$t1,5  mul $t5,$t3,3  add $t4,$t4,$t5  sub $t4,$t3,$t4  termination:  li $v0,4  la $a0,apot  syscall  li $v0,1  move $a0,$t4  syscall  li $v0,10  syscall  .data  mes: .asciiz "give number:"  apot: .asciiz "result=" | **Step 6**  .text 0x00400000  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $t1,$v0  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $t2,$v0  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $t3,$v0  # first calculation  add $t0,$t1,$t2  sub $t0,$t0,$t3  bltz $t0,less\_zero  bgt $t0,5,range  beq $t0,5,eq\_five  j termination  less\_zero:  li $t1,1  li $t2,0  start:  mul $t3,$t1,$t1  mul $t3,$t3,$t1  add $t2,$t2,$t3  add $t1,$t1,1  ble $t1,10,start  li $v0,1  move $a0,$t2  syscall  j termination  range:  blt $t0,6,termination  bgt $t0,10,termination  li $t1,10  li $t2,0  start2:  mul $t3,$t1,$t1  add $t2,$t2,$t3  add $t1,$t1,-1  bgtz $t1,start2  li $v0,1  move $a0,$t2  syscall  j termination  eq\_five:  li $v0,4  la $a0,mes2  syscall  termination:  li $v0,10  syscall  .data  mes1: .asciiz "give number:"  mes2: .asciiz "HELLO" |

|  |  |
| --- | --- |
| **Exercise 6**  **Step 1**  .text 0x00400000  li $v0,4  la $a0,m\_a  syscall  li $v0,5  syscall  move $t1,$v0  li $v0,4  la $a0,m\_b  syscall  li $v0,5  syscall  move $t2,$v0  li $v0,4  la $a0,m\_c  syscall  li $v0,5  syscall  move $t3,$v0  #calculation  mul $t4,$t2,$t2  mul $t5,$t1,4  mul $t5,$t5,$t3  sub $t0,$t4,$t5  #check  bgtz $t0,two\_roots  beqz $t0,double\_root  #imaginary  li $v0,4  la $a0,img  syscall  j termination  two\_roots:  li $v0,4  la $a0,dyo  syscall  j termination  double\_root:  li $v0,4  la $a0,dipli  syscall  termination:  li $v0,10  syscall  .data  m\_a: .asciiz "give A:"  m\_b: .asciiz "give B:"  m\_c: .asciiz "give C:"  img: .asciiz "\nMigadikes rizes"  dyo: .asciiz "\nDyo pragmatikes rizes"  dipli: .asciiz "\nMia dipli riza"  **Step 3b**  .text 0x00400000  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  #calculation  sub $t2,$t1,2  li $t4,2  div $t2,$t4  mfhi $t2  mul $t3,$t1,$t1  mul $t3,$t3,$t1  sub $t3,$t3,1  div $t0,$t2,$t3  li $v0,4  la $a0,fx  syscall  li $v0,1  move $a0,$t0  syscall  li $v0,10  syscall  .data  x: .asciiz "give X:"  fx: .asciiz "f(x)="  **Step 4a**  .text 0x00400000  start:  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  mul $t2,$t1,$t1  li $t6,2  div $t2,$t6  mfhi $t6  mul $t6,$t6,2  beqz $t6,start  #other calculations  mul $t3,$t1,$t1  mul $t3,$t3,$t1  div $t3,$t1  mfhi $t4  div $t0,$t4,$t6  li $v0,4  la $a0,fx  syscall  li $v0,1  move $a0,$t0  syscall  li $v0,10  syscall  .data  x: .asciiz "give X:"  fx: .asciiz "f(x)=" | **Step 3a**  .text 0x00400000  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  #calculation  mul $t2,$t1,$t1 #x^2  sub $t3,$t2,$t1  add $t4,$t1,2  div $t0,$t3,$t4  li $v0,4  la $a0,fx  syscall  li $v0,1  move $a0,$t0  syscall  li $v0,10  syscall  .data  x: .asciiz "give X:"  fx: .asciiz "f(x)="  **Step 3c**  .text 0x00400000  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  #calculation  mul $t3,$t1,$t1  mul $t3,$t3,$t1  div $t3,$t1  mfhi $t4  mul $t2,$t1,$t1  li $t6,2  div $t2,$t6  mfhi $t6  mul $t6,$t6,2  div $t0,$t4,$t6  li $v0,4  la $a0,fx  syscall  li $v0,1  move $a0,$t0  syscall  li $v0,10  syscall  .data  x: .asciiz "giv X:"  fx: .asciiz "f(x)="  **Step 3d**  .text 0x00400000  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  #calculation  mul $t2,$t1,$t1  mul $t3,$t2,$t1  sub $t4,$t1,3  add $t5,$t1,1  mul $t5,$t4,$t5  li $t6,1  div $t5,$t6,$t5  add $t0,$t2,$t3  sub $t0,$t0,$t5  li $v0,4  la $a0,fx  syscall  li $v0,1  move $a0,$t0  syscall  li $v0,10  syscall  .data  x: .asciiz "give X:"  fx: .asciiz "f(x)="  **Step 4b**  .text 0x00400000  start:  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  sub $t4,$t1,3  add $t5,$t1,1  mul $t5,$t4,$t5  beqz $t5,start  #other calculations  mul $t2,$t1,$t1  mul $t3,$t2,$t1  li $t6,1  div $t5,$t6,$t5  add $t0,$t2,$t3  sub $t0,$t0,$t5  li $v0,4  la $a0,fx  syscall  li $v0,1  move $a0,$t0  syscall  li $v0,10  syscall  .data  x: .asciiz "give X:"  fx: .asciiz "f(x)=" |

**Exercise 7**

|  |  |
| --- | --- |
| **Step 5**  .text 0x00400000  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  li $t2,2  start:  div $t1,$t2  mflo $t1  mfhi $t3  li $v0,1  move $a0,$t3  syscall  bgtz $t1,start  li $v0,10  syscall  .data  x: .asciiz "give X:" | **Step 7**  .text 0x00400000  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  li $t0,1  start:  li $t2,1  start2:  li $v0,4  la $a0,asterisk  syscall  add $t2,$t2,1  ble $t2,$t0,start2  li $v0,4  la $a0,line  syscall  add $t0,$t0,1  ble $t0,$t1,start  li $v0,10  syscall  .data  x: .asciiz "give N:"  line: .asciiz "\n"  asterisk: .asciiz "\*" |

**Exercise 8**

|  |  |
| --- | --- |
| **Step 10**  .text 0x00400000  li $t2,1  li $t1,0  start:  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start  li $t2,1  li $t1,0  start2:  lw $t0,arrayA($t1)  move $a0,$t0  li $v0,1  syscall  li $v0,4  la $a0,space\_char  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  li $v0,10  syscall  .data  .align 2  arrayA: .space 40  space\_char: .asciiz " " | **Step 13**  .text 0x00400000  li $t2,1  li $t1,0  start:  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start  li $t2,1  li $t1,0  start2:  lw $t0,arrayA($t1)  move $a0,$t0  li $v0,1  syscall  li $v0,4  la $a0,space\_char  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  li $v0,10  syscall  .data  .align 2  arrayA: .space 40  space\_char: .asciiz " "  mes1: .asciiz "A["  mes2: .asciiz "]=" |

**Exercise 9**

|  |  |
| --- | --- |
| **Step 2**  .text 0x00400000  li $t2,1  li $t1,0  start:  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start  li $t2,1  li $t1,0  start2:  lw $t0,arrayA($t1)  blez $t0,syn  mul $t0,$t0,$t0  sw $t0,arrayA($t1)  lw $t0,arrayA($t1)  syn:  move $a0,$t0  li $v0,1  syscall  li $v0,4  la $a0,space\_char  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  li $v0,10  syscall  .data  .align 2  arrayA: .space 40  space\_char: .asciiz " "  mes1: .asciiz "A["  mes2: .asciiz "]="  **Step 5**  .text 0x00400000  li $t2,1  li $t1,0  start:  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start  li $t2,1 #loop  li $t1,0 #deviation  li $t3,0 #sum  li $t4,2 #mod  li $t7,0 #odd numbers  start2:  lw $t0,arrayA($t1)  div $t0,$t4  mfhi $t5  beqz $t5,even  add $t7,$t7,1  j syn  even:  add $t3,$t3,$t0  syn:  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  li $v0,4  la $a0,mes3  syscall  move $a0,$t3  li $v0,1  syscall  li $v0,4  la $a0,mes4  syscall  move $a0,$t7  li $v0,1  syscall  li $v0,10  syscall  .data  .align 2  arrayA: .space 40  space\_char: .asciiz " "  mes1: .asciiz "A["  mes2: .asciiz "]="  mes3: .asciiz "SUM (even num)="  mes4: .asciiz "\nodd numbers="  **Step 8**  .text 0x00400000  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  li $t2,2  li $t0,0  start:  div $t1,$t2  mflo $t1  mfhi $t3  sw $t3,array($t0)  add $t0,$t0,4  bgtz $t1,start  #display  li $t0,36  start2:  lw $t3,array($t0)  li $v0,1  move $a0,$t3  syscall  add $t0,$t0,-4  bgez $t0,start2  li $v0,10  syscall  .data  x: .asciiz "give X:"  .align 2  array: .space 40  **Step 9**  .text 0x00400000  li $v0,4  la $a0,x  syscall  li $v0,5  syscall  move $t1,$v0  li $t2,2  li $t0,0  start:  div $t1,$t2  mflo $t1  mfhi $t3  sw $t3,array($t0)  add $t0,$t0,4  bgtz $t1,start  #display  add $t0,$t0,-4  start2:  lw $t3,array($t0)  li $v0,1  move $a0,$t3  syscall  add $t0,$t0,-4  bgez $t0,start2  li $v0,10  syscall  .data  x: .asciiz "give X:"  .align 2  array: .space 40 | **Step 3**  .text 0x00400000  li $t2,1  li $t1,0  start:  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start  li $t2,1  li $t1,0  li $t3,0  start2:  lw $t0,arrayA($t1)  add $t3,$t3,$t0  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  li $v0,4  la $a0,mes3  syscall  move $a0,$t3  li $v0,1  syscall  li $v0,10  syscall  .data  .align 2  arrayA: .space 40  space\_char: .asciiz " "  mes1: .asciiz "A["  mes2: .asciiz "]="  mes3: .asciiz "SUM="  **Step 6**  .text 0x00400000  li $t2,1  li $t1,0  start:  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start  li $t2,1 #loop  li $t1,0 #deviation  li $t3,0 #sum  li $t4,2 #mod  li $t7,0 #odd numbers  start2:  lw $t0,arrayA($t1)  div $t0,$t4  mfhi $t5  beqz $t5,even  add $t3,$t3,$t0  j syn  even:  add $t7,$t7,1  syn:  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  li $v0,4  la $a0,mes3  syscall  move $a0,$t3  li $v0,1  syscall  li $v0,4  la $a0,mes4  syscall  move $a0,$t7  li $v0,1  syscall  bgt $t7,$t3,zeros  j termination  zeros:  li $t2,1 #loop  li $t1,0 #deviation  li $t3,0 #zero numbers  start3:  lw $t0,arrayA($t1)  bnez $t0,syn2  add $t3,$t3,1  syn2:  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start3  li $v0,4  la $a0,mes5  syscall  move $a0,$t3  li $v0,1  syscall  termination:  li $v0,10  syscall  .data  .align 2  arrayA: .space 40  space\_char: .asciiz " "  mes1: .asciiz "A["  mes2: .asciiz "]="  mes3: .asciiz "SUM (odd numbers)="  mes4: .asciiz "\neven numbers="  mes5: .asciiz "\nzero numbers=" |

**Exercise 10**

|  |  |
| --- | --- |
| **Step 2**  .text 0x00400000  li $t2,1  li $t1,0  start:  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,5  syscall  move $t0,$v0  #fill array A  sw $t0,arrayA($t1)  li $v0,4  la $a0,mes3  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,5  syscall  move $t0,$v0  #fill array B  sw $t0,arrayB($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start  #display arrays before  li $t2,1  li $t1,0  start3:  lw $t4,arrayA($t1)  lw $t5,arrayB($t1)  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,1  move $a0,$t4  syscall  li $v0,4  la $a0,space\_char  syscall  li $v0,4  la $a0,mes3  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,1  move $a0,$t5  syscall  li $v0,4  la $a0,space\_char  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start3  #swap (arrayA <=> arrayB)  li $t2,1  li $t1,0  start2:  lw $t4,arrayA($t1)  lw $t5,arrayB($t1)  move $t6,$t4  sw $t5,arrayA($t1)  sw $t6,arrayB($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  li $v0,4  la $a0,line  syscall  #display arrays after  li $t2,1  li $t1,0  start4:  lw $t4,arrayA($t1)  lw $t5,arrayB($t1)  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,1  move $a0,$t4  syscall  li $v0,4  la $a0,space\_char  syscall  li $v0,4  la $a0,mes3  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,1  move $a0,$t5  syscall  li $v0,4  la $a0,space\_char  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start4  li $v0,10  syscall  .data  .align 2  arrayA: .space 40  arrayB: .space 40  space\_char: .asciiz " "  mes1: .asciiz "A["  mes2: .asciiz "]="  mes3: .asciiz "B["  line: .asciiz "\n"  **Step 6**  .text 0x00400000  li $t1,-10  again:  mul $t0,$t1,$t1  addi $t0,$t0,-1  again2:  li $v0,4  la $a0,space\_char  syscall  addi $t0,$t0,-1  bgtz $t0,again2  li $v0,4  la $a0,asterisk  syscall  addi $t1,$t1,1  ble $t1,10,again  li $v0,10  syscall  .data  asterisk: .asciiz "\*\n"  space\_char: .asciiz " "  **Step 9**  .text 0x00400000  #fill  li $t1,1  li $t3,0  start:  li $t2,1  start2:  li $v0,4  la $a0,mes1  syscall  li $v0,1  move $a0,$t1  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,1  move $a0,$t2  syscall  li $v0,4  la $a0,mes3  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,array($t3)  add $t3,$t3,4  add $t2,$t2,1  ble $t2,5,start2  add $t1,$t1,1  ble $t1,5,start  #display array  li $t1,1  li $t3,0  start3:  li $t2,1  start4:  lw $t0,array($t3)  li $v0,1  move $a0,$t0  syscall  li $v0,4  la $a0,space\_char  syscall  add $t3,$t3,4  add $t2,$t2,1  ble $t2,5,start4  li $v0,4  la $a0,line  syscall  add $t1,$t1,1  ble $t1,5,start3  #display diagonal (scan all the array)  li $v0,4  la $a0,diag  syscall  li $t1,1  li $t3,0  start5:  li $t2,1  start6:  lw $t0,array($t3)  bne $t2,$t1,cont  li $v0,1  move $a0,$t0  syscall  li $v0,4  la $a0,space\_char  syscall  cont:  add $t3,$t3,4  add $t2,$t2,1  ble $t2,5,start6  #li $v0,4  #la $a0,line  #syscall  add $t1,$t1,1  ble $t1,5,start5  li $v0,10  syscall  .data  mes1: .asciiz "A["  mes2: .asciiz ","  mes3: .asciiz "]="  space\_char: .asciiz " "  line: .asciiz "\n"  diag: .asciiz "\n\n\n \*\*\* diagonal"  .align 2  array: .space 40 | **Step 3**  .text 0x00400000  li $t2,1  li $t1,0  start:  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start  #swap  li $t2,1  li $t1,0  li $t3,20  start2:  lw $t0,arrayA($t1)  lw $t4,arrayA($t3)  move $t5,$t0  sw $t4,arrayA($t1)  sw $t5,arrayA($t3)  addi $t1,$t1,4  addi $t3,$t3,4  addi $t2,$t2,1  ble $t2,5,start2  #after  li $t2,1  li $t1,0  start3:  lw $t0,arrayA($t1)  li $v0,1  move $a0,$t0  syscall  li $v0,4  la $a0,space\_char  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start3  li $v0,10  syscall  .data  .align 2  arrayA: .space 40  space\_char: .asciiz " "  **Step 4**  .text 0x00400000  li $t2,1 #  li $t1,0  again:  li $v0,4  la $a0,mes1  syscall  move $a0,$t2  li $v0,1  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,pinax($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,again  li $t2,1  li $t1,0  again2:  lw $t3,pinax($t1)  li $v0,4  la $a0,mes3  syscall  move $a0,$t2  li $v0,1  syscall  li $v0,4  la $a0,mes4  syscall  bgtz $t3,display  return:  la $a0,new\_line  li $v0,4  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,again2  li $v0,10  syscall  display:  move $t4,$t3  again3:  la $a0,asterisk  li $v0,4  syscall  sub $t4,$t4,1  bgtz $t4,again3  j return  .data  mes1: .asciiz "A["  mes2: .asciiz "]="  mes3: .asciiz "["  mes4: .asciiz "]"  new\_line: .asciiz "\n"  asterisk: .asciiz "\*"  space\_char: .asciiz " "  .align 2  pinax: .space 40  **Step 5**  .text 0x00400000  main:  li $t1,-10  again:  mul $t0,$t1,$t1  again2:  li $v0,4  la $a0,asterisk  syscall  addi $t0,$t0,-1  bgtz $t0,again2  li $v0,4  la $a0,line  syscall  addi $t1,$t1,1  ble $t1,10,again  li $v0,10  syscall  .data  asterisk: .asciiz "\*"  line: .asciiz "\n" |

**Exercise 11**

|  |  |
| --- | --- |
| **Step 2**  .text 0x00400000  start:  jal read\_choice  move $t0,$s0  bltz $t0,start  bgt $t0,3,start  beq $t0,1,choice\_1  beq $t0,2,choice\_2  beq $t0,3,choice\_3  beqz $t0,termination  return:  j start  read\_choice:  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $s0,$v0  jr $31  choice\_1:  li $v0,4  la $a0,e1  syscall  j return  choice\_2:  li $v0,4  la $a0,e2  syscall  j return  choice\_3:  li $v0,4  la $a0,e3  syscall  j return  termination:  li $v0,10  syscall  .data  e1: .asciiz "\nchoice activated1"  e2: .asciiz "\nchoice activated2"  e3: .asciiz "\nchoice activated3"  mes1: .asciiz "\nepilogh [1-3 or 0]:"  **Step 4**  .text 0x00400000  start:  jal print\_menu  jal read\_choice  move $t0,$s0  bltz $t0,start  bgt $t0,4,start  beqz $t0,termination  jal read\_numbers  beq $t0,1,choice\_1  beq $t0,2,choice\_2  beq $t0,3,choice\_3  beq $t0,4,choice\_4  return:  jal print\_result  j start  read\_choice:  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $s0,$v0  jr $31  read\_numbers:  li $v0,4  la $a0,give\_a  syscall  li $v0,5  syscall  move $t1,$v0  li $v0,4  la $a0,give\_b  syscall  li $v0,5  syscall  move $t2,$v0  jr $31  print\_menu:  li $v0,4  la $a0,menu1  syscall  jr $31  choice\_1:  add $t3,$t1,$t2  j return  choice\_2:  sub $t3,$t1,$t2  j return  choice\_3:  mul $t3,$t1,$t2  j return  choice\_4:  div $t3,$t1,$t2  j return  print\_result:  li $v0,1  move $a0,$t3  syscall  jr $31  termination:  li $v0,10  syscall  .data  e1: .asciiz "\n \*\*\*[+]"  e2: .asciiz "\n \*\*\*[-]"  e3: .asciiz "\n \*\*\*[\*]"  e4: .asciiz "\n \*\*\*[/]"  mes1: .asciiz "\nepilogh [1-3 or 0]:"  menu1: .ascii "\n==========="  menu2: .ascii "\n1. Addition"  menu3: .ascii "\n2. Subtraction"  menu4: .ascii "\n3. Multiplication"  menu5: .ascii "\n4. Division"  menu6: .ascii "\n0. Exits"  menu7: .asciiz "\n==========="  give\_a: .asciiz "A="  give\_b: .asciiz "B=" | **Step 3**  .text 0x00400000  start:  jal print\_menu  jal read\_choice  move $t0,$s0  bltz $t0,start  bgt $t0,3,start  beq $t0,1,choice\_1  beq $t0,2,choice\_2  beq $t0,3,choice\_3  beqz $t0,termination  return:  j start  read\_choice:  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $s0,$v0  jr $31  print\_menu:  li $v0,4  la $a0,menu1  syscall  jr $31  choice\_1:  li $v0,4  la $a0,e1  syscall  j return  choice\_2:  li $v0,4  la $a0,e2  syscall  j return  choice\_3:  li $v0,4  la $a0,e3  syscall  j return  termination:  li $v0,10  syscall  .data  e1: .asciiz "\n \*\*\*choice activated1"  e2: .asciiz "\n \*\*\*choice activated2"  e3: .asciiz "\n \*\*\*choice activated3"  mes1: .asciiz "\nepilogh [1-3 or 0]:"  menu1: .ascii "\n==========="  menu2: .ascii "\n1. item 1"  menu3: .ascii "\n2. item 2"  menu4: .ascii "\n3. item 3"  menu5: .ascii "\n0. Exodos"  menu6: .asciiz "\n==========="  **Step 5**  .text 0x00400000  li $t6,0 #stack pointer  start:  jal print\_menu  jal read\_choice  move $t0,$s0  bltz $t0,start  bgt $t0,3,start  beq $t0,1,choice\_1  beq $t0,2,choice\_2  beq $t0,3,choice\_3  beqz $t0,termination  return:  jal print\_stack  j start  read\_choice:  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $s0,$v0  jr $31  print\_menu:  li $v0,4  la $a0,menu1  syscall  jr $31  choice\_1:  li $t3,1  li $t4,0  li $t5,0  again:  sw $t5,stack($t4)  add $t4,$t4,4  add $t3,$t3,1  ble $t3,10,again  li $t6,0  j return  choice\_2:  beq $t6,10,return1  li $v0,4  la $a0,timh  syscall  li $v0,5  syscall  move $t7,$v0  add $t6,$t6,1  mul $t3,$t6,4  sub $t3,$t3,4  #add $t6,$t6,1  sw $t7,stack($t3)  #j return4  return1:  #li $t6,10  #return4:  j return  choice\_3:  li $t7,0  mul $t3,$t6,4  sub $t3,$t3,4  sw $t7,stack($t3)  beq $t6,0,return2  add $t6,$t6,-1  return2:  j return  print\_stack:  li $t3,1  li $t4,0  loop:  lw $t5,stack($t4)  bnez $t5,syn  li $v0,4  la $a0,star  syscall  j syn2  syn:  li $v0,1  move $a0,$t5  syscall  syn2:  beq $t6,$t3,print\_pointer  return:  li $v0,4  la $a0,line  syscall  add $t3,$t3,1  add $t4,$t4,4  ble $t3,10,loop  jr $31  print\_pointer:  li $v0,4  la $a0,pointer  syscall  j return  termination:  li $v0,10  syscall  .data  mes1: .asciiz "\nSelect [1-3 or 0]:"  menu1: .ascii "\n==========="  menu2: .ascii "\n1. Initialization"  menu3: .ascii "\n2. Push"  menu4: .ascii "\n3. Pop"  menu5: .ascii "\n0. Exit"  menu6: .asciiz "\n==========="  line: .asciiz "\n"  star: .asciiz "\*"  pointer: .asciiz "<=="  timh: .asciiz "New value="  .align 2  stack: .space 40 |

**Exercise 12**

|  |  |
| --- | --- |
| **Step 1**  .text 0x00400000  start:  jal print\_menu  jal read\_choice  move $t0,$s0  bltz $t0,start  bgt $t0,2,start  beq $t0,1,choice\_1  beq $t0,2,choice\_2  beqz $t0,termination  return:  j start  read\_choice:  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $s0,$v0  jr $31  print\_menu:  li $v0,4  la $a0,menu1  syscall  jr $31  choice\_1:  li $v0,4  la $a0,e1  syscall  li $t2,1  li $t1,0  start1:  li $v0,4  la $a0,e3  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start1  j return  choice\_2:  li $v0,4  la $a0,e2  syscall  li $t2,1  li $t1,0  start2:  lw $t0,arrayA($t1)  move $a0,$t0  li $v0,1  syscall  li $v0,4  la $a0,space\_char  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  j return  termination:  li $v0,10  syscall  .data  e1: .asciiz "\n \*\*\* Fill array (after each number press <enter>)\n"  e2: .asciiz "\n \*\*\* Display array\n"  e3: .asciiz ">"  mes1: .asciiz "\nSelect [1-2 or 0]:"  menu1: .ascii "\n==========="  menu2: .ascii "\n1. Fill"  menu3: .ascii "\n2. Display"  menu5: .ascii "\n0. Exit"  menu6: .asciiz "\n==========="  .align 2  arrayA: .space 40  space\_char: .asciiz " "  **Step 4**  .text 0x00400000  start:  jal print\_menu  jal read\_choice  move $t0,$s0  bltz $t0,start  bgt $t0,7,start  beq $t0,1,choice\_1  beq $t0,2,choice\_2  beq $t0,3,choice\_3  beq $t0,4,choice\_4  beq $t0,5,choice\_5  beq $t0,6,choice\_6  beq $t0,7,choice\_7  beqz $t0,termination  return:  j start  read\_choice:  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $s0,$v0  jr $31  print\_menu:  li $v0,4  la $a0,menu1  syscall  jr $31  choice\_1:  li $v0,4  la $a0,e1  syscall  li $t2,1  li $t1,0  start1:  li $v0,4  la $a0,e3  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start1  j return  choice\_2:  li $v0,4  la $a0,e2  syscall  li $t2,1  li $t1,0  start2:  lw $t0,arrayA($t1)  move $a0,$t0  li $v0,1  syscall  li $v0,4  la $a0,space\_char  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  j return  choice\_3:  li $v0,4  la $a0,e2  syscall  li $t2,1  li $t1,0  lw $t3,arrayA($t1)  start3:  lw $t0,arrayA($t1)  bge $t0,$t3,syn2  move $t3,$t0  syn2:  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start3  li $v0,4  la $a0,e4  syscall  li $v0,1  move $a0,$t3  syscall  j return  choice\_4:  li $v0,4  la $a0,e2  syscall  li $t2,1  li $t1,0  lw $t3,arrayA($t1)  start4:  lw $t0,arrayA($t1)  ble $t0,$t3,syn3  move $t3,$t0  syn3:  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start4  li $v0,4  la $a0,e5  syscall  li $v0,1  move $a0,$t3  syscall  j return  #Sum  choice\_5:  li $t2,1  li $t1,0  li $t3,0  start5:  lw $t0,arrayA($t1)  add $t3,$t3,$t0  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start5  li $v0,4  la $a0,e6  syscall  li $v0,1  move $a0,$t3  syscall  j return  #Zero, positive, negative  choice\_6:  li $t2,1  li $t1,0  li $t4,0 #positive  li $t5,0 #negative  li $t6,0 #zeros  start6:  lw $t0,arrayA($t1)  beqz $t0,zero  bgtz $t0,meg  add $t5,$t5,1  syn4:  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start6  li $v0,4  la $a0,e7  syscall  li $v0,1  move $a0,$t6  syscall  li $v0,4  la $a0,e8  syscall  li $v0,1  move $a0,$t4  syscall  li $v0,4  la $a0,e9  syscall  li $v0,1  move $a0,$t5  syscall  j return  zero:  add $t6,$t6,1  j syn4  meg:  add $t4,$t4,1  j syn4  #Sorting  choice\_7:  #Homework!!!  j return  termination:  li $v0,10  syscall  .data  e1: .asciiz "\n \*\*\* Fill array (after each number press <enter>)\n"  e2: .asciiz "\n \*\*\* Display array\n"  e3: .asciiz ">"  e4: .asciiz "\nMin="  e5: .asciiz "\nMax="  e6: .asciiz "\nSum="  e7: .asciiz "\n Zeros="  e8: .asciiz "\n Positive="  e9: .asciiz "\n Negative="  mes1: .asciiz "\nSelect [1-4 or 0]:"  menu1: .ascii "\n==========="  menu2: .ascii "\n1. Fill"  menu3: .ascii "\n2. Display"  menu4: .ascii "\n3. Find min"  menu5: .ascii "\n4. Find max"  menu6: .ascii "\n5. Sum"  menu7: .ascii "\n6. Zeros, positive, negative"  manu8: .ascii "\n7. Sorting"  menu9: .ascii "\n0. Exit"  menu10: .asciiz "\n==========="  .align 2  arrayA: .space 40  space\_char: .asciiz " " | **Step 3**  .text 0x00400000  start:  jal print\_menu  jal read\_choice  move $t0,$s0  bltz $t0,start  bgt $t0,4,start  beq $t0,1,choice\_1  beq $t0,2,choice\_2  beq $t0,3,choice\_3  beq $t0,4,choice\_4  beqz $t0,termination  return:  j start  read\_choice:  li $v0,4  la $a0,mes1  syscall  li $v0,5  syscall  move $s0,$v0  jr $31  print\_menu:  li $v0,4  la $a0,menu1  syscall  jr $31  choice\_1:  li $v0,4  la $a0,e1  syscall  li $t2,1  li $t1,0  start1:  li $v0,4  la $a0,e3  syscall  li $v0,5  syscall  move $t0,$v0  sw $t0,arrayA($t1)  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start1  j return  choice\_2:  li $v0,4  la $a0,e2  syscall  li $t2,1  li $t1,0  start2:  lw $t0,arrayA($t1)  move $a0,$t0  li $v0,1  syscall  li $v0,4  la $a0,space\_char  syscall  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start2  j return  choice\_3:  li $v0,4  la $a0,e2  syscall  li $t2,1  li $t1,0  lw $t3,arrayA($t1)  start3:  lw $t0,arrayA($t1)  bge $t0,$t3,syn2  move $t3,$t0  syn2:  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start3  li $v0,4  la $a0,e4  syscall  li $v0,1  move $a0,$t3  syscall  j return  choice\_4:  li $v0,4  la $a0,e2  syscall  li $t2,1  li $t1,0  lw $t3,arrayA($t1)  start4:  lw $t0,arrayA($t1)  ble $t0,$t3,syn3  move $t3,$t0  syn3:  addi $t1,$t1,4  addi $t2,$t2,1  ble $t2,10,start4  li $v0,4  la $a0,e5  syscall  li $v0,1  move $a0,$t3  syscall  j return  termination:  li $v0,10  syscall  .data  e1: .asciiz "\n \*\*\* Fill array (after each number press <enter>)\n"  e2: .asciiz "\n \*\*\* Display array\n"  e3: .asciiz ">"  e4: .asciiz "\nMin="  e5: .asciiz "\nMax="  mes1: .asciiz "\nSelect [1-4 or 0]:"  menu1: .ascii "\n==========="  menu2: .ascii "\n1. Fill"  menu3: .ascii "\n2. Display"  menu4: .ascii "\n3. Find min"  menu5: .ascii "\n4. Find max"  menu6: .ascii "\n0. Exit"  menu7: .asciiz "\n==========="  .align 2  arrayA: .space 40  space\_char: .asciiz " " |

**Exercise 13**

|  |  |
| --- | --- |
| **Step 1**  .text 0x00400000  li $v0,4  la $a0,mes  syscall  li $v0,8  la $a0,alpha  li $a1,11  syscall  li $v0,10  syscall  .data  alpha: .space 11  mes: .asciiz "String:"  **Step 5**  .text 0x00400000  la $a0,mes\_word  li $v0,4  syscall  la $a0,word  li $a1,31  li $v0,8  syscall  la $t0,word  again:  lb $t1,($t0)  beqz $t1,again\_exit  blt $t1,'a',no\_change  bgt $t1,'z',no\_change  addi $t1,$t1,-32  no\_change:  sb $t1,($t0)  addi $t0,$t0,1  j again  again\_exit:  la $a0,mes\_capitals  li $v0,4  syscall  la $a0,word  li $v0,4  syscall  li $v0,10  syscall  .data  mes\_word: .asciiz "Word (small letters):"  mes\_capitals: .asciiz "\nword with capitals:"  word: .space 41  **Step 7**  .text 0x00400000  la $a0,mes\_word  li $v0,4  syscall  la $a0,word  li $a1,31  li $v0,8  syscall  la $t0,word  again:  lb $t1,($t0)  beqz $t1,again\_exit  blt $t1,'a',no\_change  bgt $t1,'z',no\_change  addi $t1,$t1,-1  no\_change:  sb $t1,($t0)  addi $t0,$t0,1  j again  again\_exit:  la $a0,mes\_capitals  li $v0,4  syscall  la $a0,word  li $v0,4  syscall  li $v0,10  syscall  .data  mes\_word: .asciiz "Word (encrypted):"  mes\_capitals: .asciiz "\nInitial word:"  word: .space 21  **Step 9**  .text 0x00400000  la $a0,mes\_word  li $v0,4  syscall  la $a0,word  li $a1,31  li $v0,8  syscall  la $t0,word  li $t3,0  li $t5,0  again:  lb $t1,($t0)  beqz $t1,again\_exit  blt $t1,'a',no\_change  bgt $t1,'z',no\_change  beq $t1,'a',char\_a  beq $t1,'b',char\_b  no\_change:  sb $t1,($t0)  addi $t0,$t0,1  j again  again\_exit:  beqz $t3,syn2  la $a0,fa  li $v0,4  syscall  move $t4,$t3  jal print  syn2:  beqz $t5,syn3  la $a0,fb  li $v0,4  syscall  move $t4,$t5  jal print  syn3:  li $v0,10  syscall  char\_a:  add $t3,$t3,1  j no\_change  char\_b:  add $t5,$t5,1  j no\_change  print:  li $t6,1  start:  li $v0,4  la $a0,star  syscall  add $t6,$t6,1  ble $t6,$t4,start  jr $31  .data  mes\_word: .asciiz "Word:"  fa: .asciiz "\n[a]"  fb: .asciiz "\n[b]"  star: .asciiz "\*"  word: .space 31 | **Step 2**  .text 0x00400000  li $v0,4  la $a0,mes1  syscall  li $v0,8  la $a0,alpha  li $a1,11  syscall  li $v0,4  la $a0,mes2  syscall  li $v0,4  la $a0,alpha  syscall  li $v0,10  syscall  .data  alpha: .space 11  mes1: .asciiz "String:"  mes2: .asciiz "\n\nThe string is:"  **Step 6**  .text 0x00400000  la $a0,mes\_word  li $v0,4  syscall  la $a0,word  li $a1,31  li $v0,8  syscall  la $t0,word  again:  lb $t1,($t0)  beqz $t1,again\_exit  blt $t1,'a',no\_change  bgt $t1,'z',no\_change  addi $t1,$t1,1  no\_change:  sb $t1,($t0)  addi $t0,$t0,1  j again  again\_exit:  la $a0,mes\_capitals  li $v0,4  syscall  la $a0,word  li $v0,4  syscall  li $v0,10  syscall  .data  mes\_word: .asciiz " Word (small letters):"  mes\_capitals: .asciiz "\nEncrypted word"  word: .space 21  **Step 8**  .text 0x00400000  la $a0,mes\_word  li $v0,4  syscall  la $a0,word  li $a1,31  li $v0,8  syscall  la $t0,word  li $t3,0  again:  lb $t1,($t0)  beqz $t1,again\_exit  blt $t1,'a',no\_change  bgt $t1,'z',no\_change  bne $t1,'a', no\_change  add $t3,$t3,1  no\_change:  sb $t1,($t0)  addi $t0,$t0,1  j again  again\_exit:  la $a0,frequency  li $v0,4  syscall  li $v0,1  move $a0,$t3  syscall  li $v0,10  syscall  .data  mes\_word: .asciiz "Word:"  frequency: .asciiz "\nNumber of symbols 'a'="  word: .space 21 |