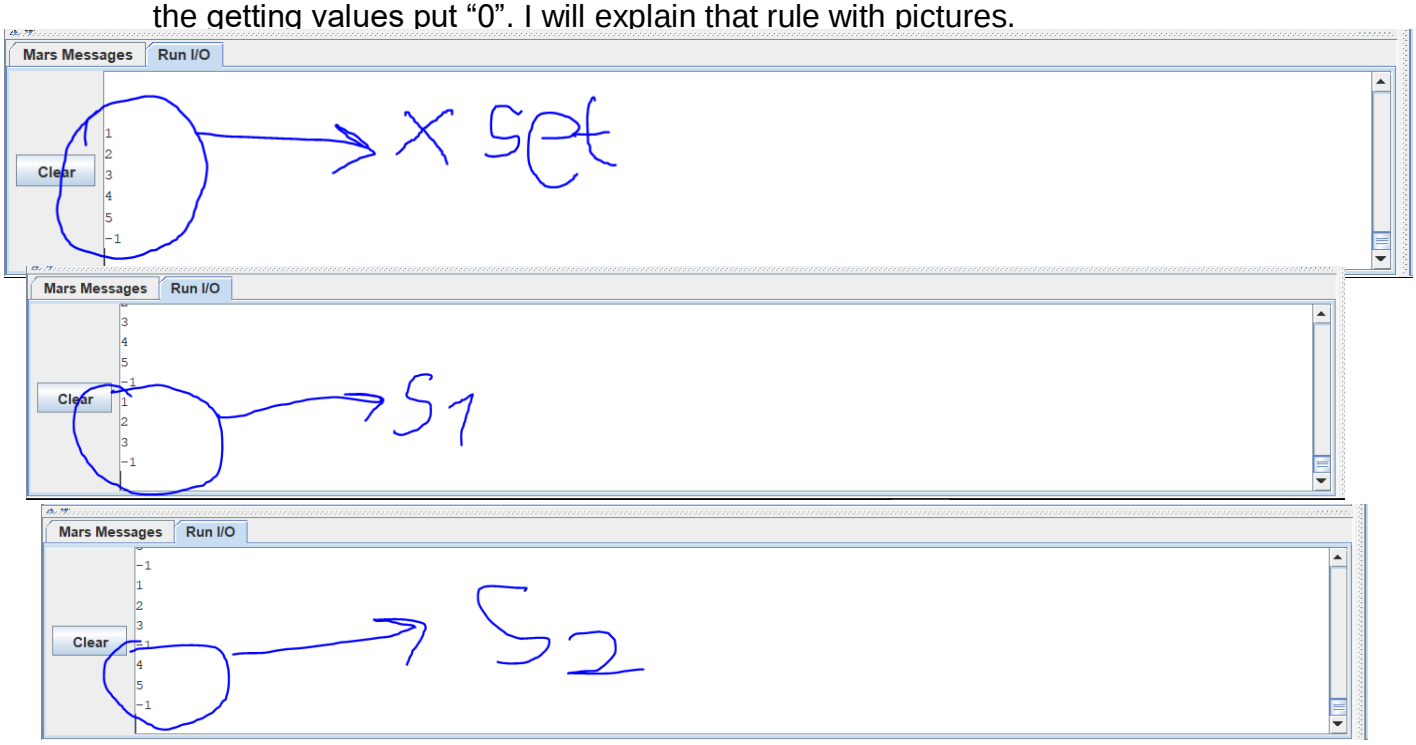
CSE 331 COMPUTER ORGANIZATION

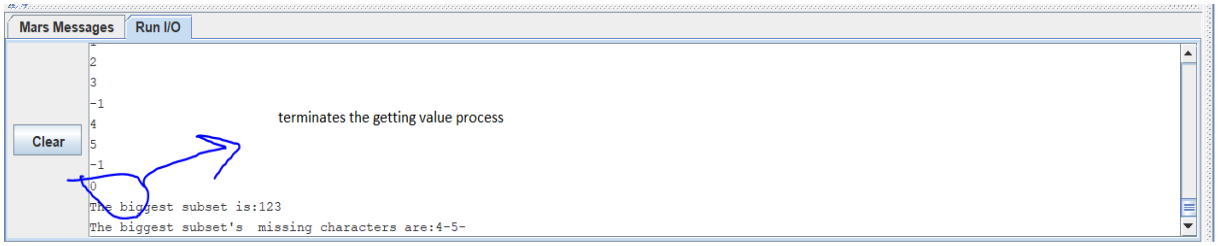
#HW1

ALP EMİR BİLEK

161044049

First of all my program getting the all sets from user via console.There is a rule for the getting the sets that you should put “-1” after the every single set and terminating the getting values put “0”. I will explain that rule with pictures.





After the explaining my rules let’s start to algrotihm.

I have 3 arrays fort he sets.First array's name is arr.Arr for the X set.

I store the whole subset in arr2.

```
loop:
        li $v0,5
        syscall
        move $s0,$v0
        beq $s0,-1,end
        sw $s0,arr($t0)
        addi $t0,$t0,4
        j loop
end:
        move $s4,$t0
        addi $s1,$zero,0
        addi $t0,$zero,0
        loop3:
```

$t0 is a iterator fort he arr.

$s0 is used fort he assigning to arr.

If you enter -1 arr will be done.

$t0 is a iterator again

$s1 is used fort he assigning to arr2

If you enter 0 arr2 will be done

If you enter -1 arr2 will store the next subset.

And finding the biggest subset process will start.

```
    loop3:
        li $v0,5
        syscall
        move $s1,$v0
        beq $s1,0,end1
        beq $s1,-1,end2
        sw $s1,arr2($t0)
        addi $t0,$t0,4
        addi $t2,$t2,4
        addi $t3,$t3,4
        j loop3

    end1:
        lw $t5,arr2($zero)
        addi $t0,$zero,0
        addi $t7,$zero,0
        add $t7,$t4,$t1
```

$t1's first value is 0 and we comparing the $t1 and $t2.

$t2 is a temp value.$t2 counts the indexes between two -1

If $t2 is bigger than $t1 we swap two registers.

Swap is a procedure that $t2 moves to $t1 and $t2 will 0.

We hold the location of the biggest array's head.

If $t2 is smaller than $t1 we make $t2 0 via zer procedure.

```
    end2:
        blt $t1,$t2,s1
        bgt $t1,$t2,s2
    s1:
        jal swap
        j loop3
    s2:
        jal zer
        j loop3
    swap:
        move $t1,$t2
        sub $t7,$t3,$t1
        move $t4,$t7
        addi $t2,$zero,0
        jr $ra
    zer:
        addi $t2,$zero,0
        jr $ra
```

```
end1:
        lw $t5,arr2($zero)
        addi $t0,$zero,0
        addi $t7,$zero,0
        add $t7,$t4,$t1
loop4:
        beq $t4,$t7,end3
        lw $t6,arr2($t4)
        addi $t4,$t4,4
        sw $t6,arr3($t0)
        addi $t0,$t0,4
        j loop4
end3:
        li $v0,4
        la $a0,bigarr
        syscall
        addi $t7,$zero,0
        loop78:
                beq $t7,$t1,exit12
                lw $s6,arr3($t7)
                li $v0,1
                move $a0,$s6
                syscall
                addi $t7,$t7,4
                j loop78
        exit12:
```

After finding the biggest array.We push it to arr3.

```
Search:
        li $v0,4
        la $a0,newline
        syscall
        li $v0,4
        la $a0,missing
        syscall
        addi $t0,$zero,0
        addi $t1,$zero,0
        move $t0,$s4
        move $t1,$s7
        addi $t0,$t0,4
        addi $t2,$zero,0
        addi $t3,$zero,0
        addi $t4,$zero,4
        MainLoop:
                beq $t2,$t0,endMain
                beq $t4,4,go_somewhere
                beq $t4,0,push
                InnerLoop:
                        beq $t3,$t1,endInner

                        lw $t6,arr($t2)
                        lw $t5,arr3($t3)

                        beq $t5,$t6,inc

                        addi $t3,$t3,4
```

After pushing process we start the search process.

Search process searches the different values between

2 arrays.

We have 2 loops for the search.First one is for X set.

Second one is for the biggest subset.Those loops are

Too similar to for loops in C programming.

The algorithm is if two arrays values equal,it will jump to inc.

In the inc part we call increase procedure.

In that procedure we increase $t2(iterator for the x set)

We make 0 $t3(iterator for the biggest subset)

```
inc:
        jal Increase
        j MainLoop

Increase:
        addi $t4,$t4,4
        addi $t2,$t2,4
        addi $t3,$zero,0
        jr $ra

endInner:
        jal ender
        j MainLoop

ender:
        addi $t3,$zero,0
        addi $t2,$t2,4
        jr $ra

go_somewhere:
        jal do_something
        j InnerLoop
do_something:
        addi $t4,$zero,0
        jr  $ra
push:
        jal push_f
        j InnerLoop
push_f:
        addi $t4,$zero,0
        addi $t7,$zero,0
        subi $t7,$t2,4
        lw $s5,arr($t7)
        li $v0,1
        move $a0,$s5
        syscall
```

We increase the $t4 because checking fort he if biggest subset has that value or not.

If $t4 is 4, biggest subset has that value

Otherwise has not.

If $t4 is 0 that means we will search in the whole subset.