"STEP-WISE EXPILANATION OF THE PROJECT"

PROJECT: I want to create a pattern based on the numbers stored in the array.

```
.data
        myarr: .space 24
        newline: .asciiz "\n"
        star: .asciiz "* "
.text
                li $s0,3                # Stores the immediate value 1 in $s0
                li $s1,6                # Stores the immediate value 2 in $s1
                li $s2,9                # Stores the immediate value 3 in $s2
                li $s3,12               # Stores the immediate value 4 in $s3
                li $s4,16               # Stores the immediate value 5 in $s4
                li $s5,20               # Stores the immediate value 6 in $s5


                li $t0,0                # Stores the immediate value 0 in $t0.

                sw $s0, myarr($t0)      # Stores the integers from Register $s0 into the array.
                        addi $t0,$t0,4  # Increments the address of the array: array[$t0+4].

                sw $s1, myarr($t0)      # Stores the integers from Register $s0 into the array.
                        addi $t0,$t0,4  # Increments the address of the array: array[$t0+4].

                sw $s2, myarr($t0)      # Stores the integers from Register $s0 into the array.
                        addi $t0,$t0,4  # Increments the address of the array: array[$t0+4].
```

```
        sw $s3, myarr($t0)      # Stores the integers from Register $s0 into the array.
            addi $t0,$t0,4      # Increments the address of the array: array[$t0+4].

        sw $s4, myarr($t0)      # Stores the integers from Register $s0 into the array.
            addi $t0,$t0,4      # Increments the address of the array: array[$t0+4].

        sw $s5, myarr($t0)      # Stores the integers from Register $s0 into the array.

        addi $t0,$zero,0        # Stores the immediate value 0 in $t0.
        li $t1,0                # Stores the immediate value 0 in $t1.
        li $t2,6                # Stores the immediate value 6 in $t2.

Outerloop:
        bge  $t1,$t2,exit       # if $t1(first element) >= $t2(last element): The number
                                # wasn't found and return to exit.
        addi $t1,$t1,1          # Increments the value of $t1 Register by 1: $t1++

        lw $t4, myarr($t0)      # Loads the integer into $t4: $t4 = array[x].
            addi $t0,$t0,4      # Increments the address of the array: array[$t0+4].

        li $t3,0                # Stores the immediate value 0 in $t3.


        li $v0, 4               # System call code to print a string.
la $a0, newline                 # Loads the address of newline into the argument
                                # register.
syscall                         # Print newline.
```

```
Innerloop:
        bge  $t3,$t4,Outerloop      # if $t3(first element) >= $t4(last element): The
                                    # number wasn't found and return to Outerloop
                addi $t3,$t3,1      # Increments the value of $t3 Register by 1:  $t1++


                li $v0, 4           # System call code to print a string
          la $a0, star              # Loads the address of star into the argument register.
                syscall             # Print star


                j Innerloop         # else: Innerloop
exit:
          li $v0, 10                # System call code to exit the program.
          syscall                   # Exit the program.
```