Objective of the Lab/Program

This program calculates $^{n}C_{r,}$ in that $^{n}C_{r}$ = n! / ((n-r)! * r!). It will accept the numbers n and r from the user and calculate the factorial in a block. Should n < r it prints an error message and get the inputs n and r from the user again.

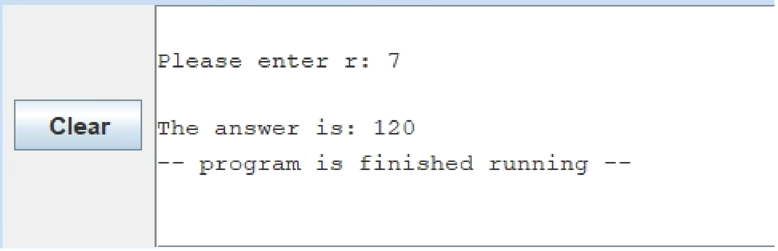
Assembly Source Codes

```
.data
str1: .asciiz "Please enter n: "
str2: .asciiz "\nPlease enter r: "
str3: .asciiz "\nn should not be less than r re-enter the values.\n"
str4: .asciiz "\nThe answer is: "
.text

j begin
again:
la $a0, str3
li $v0, 4
syscall


begin:
la $a0, str1
li $v0, 4
syscall

li $v0, 5
syscall
move $s0, $v0           #Contains the value of n


la $a0, str2
li $v0, 4
syscall

li $v0, 5
syscall
move $s1, $v0           #Contains the value of r
```

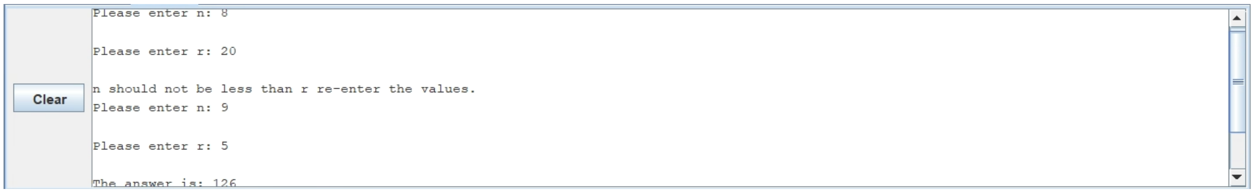
```
blt $s0, $s1, again                      #If N < R try again

#Defining Variables
move $t0, $s0                            #Holds the value of n, will also be used to
calculate factorial of a number
li $t1, 0                                #contains the value of the top of the fraction
li $t2, 0                                #Contains the value in the bottom left of the
fraction
li $t3, 0                                #Contains the value in the bottom right of
the fraction
sub $t4, $s0, $s1                        #make $t4 = n-r
li $t7, 0                                #multiply value of factorial

move $t7, $t0
jal factorial
main:
move $t1, $t0                                    #Assign the value of top of fraction
to n!

move $t0, $s1                                    #make $t0 r
move $t7, $t0                                    #make $t7 r which is then going to
be r-1....
jal factorial                                    #perform r!
move $t2, $t0

move $t0, $t4                                    #make $t0 = (n-r)
move $t7, $t0                                    #make $t7 (n-r)
jal factorial                                    #perform n-r!
move $t3, $t0

mul $t5, $t3, $t2                                #holds the finally value of the
bottom fraction
div $t1, $t5
mflo $t6


la $a0, str4                                     #Prints the result
li $v0, 4
syscall

la $a0, ($t6)
li $v0, 1
syscall

j exit
```

```
factorial:
subi $t7, $t7, 1                    #make $t7 = $t7-1
mul $t0, $t7, $t0                           #$t0 * $t7
bgt $t7, 1, factorial                       #break if $t0 > 1
jr $ra



exit:
li $v0, 10              #terminate program
syscall
```

Screen shot of the results



```
Please enter r: 7

The answer is: 120
-- program is finished running --
```



```
Please enter n: 8

Please enter r: 20

n should not be less than r re-enter the values.
Please enter n: 9

Please enter r: 5

The answer is: 126
```

Conclusion and References

A label for calculating factorial was created and whenever it was called it was treated as a function. The program individually found the factorial of three separate values, n!, r! and (n-r)! then multiplied and divided the values accordingly to arrive at the result.