# Combination and Permutation

Due: March 28, 2024

## Problem Description

Modify the RISC-V assembly programs given in https://hackmd.io/@linyu413/riscv-001 and https://hackmd.io/@linyu413/1122HW102 for computing combination and permutation number. The function of this code is to calculate the factorial number of m, but this code has obvious bugs, please try to find out where the bugs are and correct them. Please make the first line of the printed message be your student ID number.

You should use RISC-V instruction set simulator **RARS** to develop and execute the assembly code.

The RARS simulator can be downloaded from https://www.rose-hulman.edu/class/csse/csse232/Lab1/rars_27a7c1f.jar .

To understand the code in https://hackmd.io/@linyu413/riscv-001 and https://hackmd.io/@linyu413/1122HW102, you need study how the assembler, directives, ecalls, etc. are used to develop the assembly code. In particular, **ecalls** implements a set of ABI's (Application Binary Interface) that can be used by the assembly code to interact with an operating system to read data from a keyboard and write data onto a monitor.

It is very easy to use RARS. Simply download the related file in its web site. Doubly click the file to invoke RARS (need Java Run Time). Then, you can write, assemble, execute, and debug the code directly on RARS.

## Input Format

When the code is executed on RARS, Just follows the instruction presented on the monitor to provide input data.

## Output Format

The output format should be exactly the same as the example given below. Certainly, different input will result in different numbers presented on the monitor.

## What Should Be Handed In:

- Assembly code for modified https://hackmd.io/@linyu413/riscv-001 and https://hackmd.io/@linyu413/1122HW102. **The first line of assembly code should consist of your student ID number**. Every **added** instruction should have a comment to explain what the instruction does. a comment should start with ### at the beginning of the comment. The file name of the assembly code should be **sID.asm** where ID is your student ID number. A valid file name will look like s1111111.asm .
- A clip like the one shown in the example of input and output below. Save the clip as a file called **sID.png**, where ID is your student ID number. A valid file name for an output clip will look like s1111111.png .
- The homework will not be graded if you do not follow the above rules.

## Hint:

| | |
|---|---|
| 1. 排列(P) | P n 取k = n! / (n-k)! |
| 2. 組合(C) | C n 取k = n! / k! (n-k)! |
| 3. 重複排列 | $n^R$ |
| 4. 重複組合(H) | H n取k = C (n+k-1)取k |

## Grade point rules:

Basic:
Student ID (20%)
Number of permutation (20%)
Number of combination (20%)
Advanced:
Number of repeat combination (20%)
Number of repeat permutation (20%)
Extra points:
Bug fixing and fool proofing (10%)

## Example of input and output: (basic)

```
s111111
input number M=4
input number N=2
P(M,N)=12
C(M,N)=6
```