Exercises in MIPS

1 Hello world

Exercice 1. In this exercise you need to write a mips program that prints "Hello World!" and a newline using the syscall 4 (print string).

Exercice 2. This second exercise is the same as the first except that you should only use the syscall 11 (print char).

Exercice 3. Now, write a mips program that reads an integer n and then prints $2 * n$.

2 Simple loops

Exercice 4. Here you need to read an integer n and then print n times the character # followed by a newline. For instance on the input 5 you should print :

```
#####
```

Exercice 5. Here you need to read an integer n and then print a triangle made of the character #. For instance on the input 5 you should print :

```
#####
####
###
##
#
```

Exercice 6. Here you need to read an integer n and then print F_n with F_n being the sequence such that $F_0 = 0$, $F_1 = 1$ and $F_{n+2} = F_{n+1} + F_n$. For this simple exercise you need to use the following algorithm (shown in C) :

```c
int cur = 0;
int nxt = 1;
while(n>0) {
  int nxtnxt = cur+nxt ;
  cur = nxt ;
  nxt = nxtnxt ;
  n--;
}
// cur holds the result
```

For reference, $F_{10} = 55$ and $F_{20} = 6765$.

3 More complicated loops

Exercice 7. Given an integer n, we define syracuse(n) as the number $n/2$ when n is even and $3n+1$ otherwise. The goal is to repetitively print n and then replace n with syracuse(n) until n reaches 1. For instance here is the output for $n = 20$ (note that n is the input and also part of the output).

```
20
10
5
16
8
4
2
1
```

Exercice 8. Consider the infinite computation $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - ....$ We consider P_k its truncation at the k-th terms, *i.e.* $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9}$ for $k = 5$.

The goal is to print $4 \times P_k$ for all k (starting from 1, then 2, etc.; yes, this is an infinite loop).

Hint : the float functions are mostly the same as the integer ones but with `.s`.

4 Recursion

Exercice 9. Compute $n!$ using recursion. Reminder : $n! = n \times (n-1) \times \cdots \times 2 \times 1$.

Exercice 10. Compute F_n (see above) using recursion. Do not try to memoize.

Exercice 11. Read three integers, a, b and c and then print $a^c \mod b$ computed using a fast exponentiation algorithm.

Exercice 12. We define T_1 the Sierpinski triangle of size 1 as the single character # and T_{2N}, the Sierpinski triangle of size $2N$ as the combination of one T_N above the concatenation of two T_N. *Below are the triangles $T_1, T_2, T_4, T_8, T_{16}$, it should make it clearer.* You probably want to use the RAM to store the triangle before printing it...

```
N=1
#

N=2
#
##

N=4
#
##
# #
####
```

```
N=8
#
##
# #
####
#   #
##  ##
# # # #
########

N=16
#
##
# #
####
#   #
##  ##
# # # #
########
#       #
##      ##
# #     # #
####    ####
#   #   #   #
##  ##  ##  ##
# # # # # # # #
################
```