CS231 Computer Systems and Organization
Department of Computer Science
School of Engineering and Digital Sciences

Lab 7

Submit your program before the deadline.

1. The trigonometric function *cosine* has the following Taylor expansion

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} - \dots$$

Implement a procedure `cosine` in MIPS assembly language that, given a value x in radian, approximates $\cos(x)$ using the expansion above. The `cosine` procedure should call two helper procedures `power` and `factorial` to find x^i and $i!$, respectively, of which the `factorial` procedure should be implemented *recursively*.

The procedure takes the argument x in register `$f12` and returns the result in register `$f0`. All floating point numbers are single precision. Your program should prompt the user for x in degree (so your program should convert x to radian first) and print out $\cos(x)$. For example, for $x = 32$, it should output 0.8480481[1], i.e., $\cos(32) = 0.8480481$. Your program should continue prompting until the user inputs -1. You can set the last term to $\frac{x^{12}}{12!}$, as 13! may not fit into a register.

The signatures of these procedures in a high level language would look like as follows: `int factorial(int n), float power(float x, int n), float cosine(float x)`

MIPS has the following conventions for the floating-point registers:

- `$f0 - $f2` Floating point procedure results
- `$f4 - $f10` Temporary registers. Not preserved across procedure calls
- `$f12 - $f14` Floating point procedure parameters. Not preserved across procedure calls
- `$f16 - $f18` Temporary floating point registers. Not preserved across procedure calls
- `$f20 - $f30` Saved floating point values. Preserved across procedure calls

Do not forget to follow other MIPS conventions.

[1] Values for $\cos(x)$ may vary depending on the precision of π. Therefore, use $\pi = 3.1415927$ to get $\cos(32) = 0.8480481$.