Descrição das funções

* *atualizajogo*:

Sendo que o valor do início da tabela está em R1, tivemos em mente guardar o valor de R1 + 1 na pilha e depois reposicioná-lo no valor apontado por R1. Iteramos este processo 80 vezes, sendo que na última usamos o valor originado pela função *geracacto* no último valor da tabela. Sabendo que usamos o valor de R4 como um registo intermedio, seguimos a convenção do P4 e salvaguardamos o seu valor no início da função.

* *geracacto*:

A primeira parte da função consiste na tradução da função apresentada em Python no enunciado, que vai gerar números pseudo-aleatórios (trata-se de uma sucessão em que os termos se repetem eventualmente). Dos números gerados, sensivelmente 95% está abaixo de 62258, pelo que quando um número for superior, vai ser gerado um cato.

Como a altura máxima do cato é dada por uma potência de 2, em binário é representado por um número com apenas um bit 1 seguido de n bits 0. Quando subtraímos 1, o número passar a ter um 0 no lugar do 1 e os n bits restantes passam a 1 (0010 0000 -> 0001 1111). Como 1 é o elemento neutro do AND lógico e o 0 é o elemento absorvente, isto vai resultar num número que apenas vai ter valores em [0, altura -1], que é equivalente ao resto da divisão (0001 1111 & 1101 0010 = 00010010) porque vai anular toda a parte do x que for múltipla de altura e deixar o que não é intacto. Logo quando somamos 1 no fim, obtemos valores em [1, altura].

* Programa de teste:

Para iniciar o programa, introduzimos a tabela com dois valores (4) em duas posições para verificarmos se a função *atualizajogo* funciona.

De seguida preparámos os registos para a chamada da função *atualizajogo*. De forma a esta função ser executada num loop, definidos um branch infinito após o retorno a função MAIN.

Para a função *geracacto* apenas iniciamos o valor de R1 (argumento da função) com a altura máxima, cujo valor é uma potência de 2 arbitrária, uma constante do programa (16).

Para testar o código basta assemblar e correr o programa (Teste.as) no simulador do P4.