ÇANKAYA UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

CENG222 COMPUTER ORGANIZATION

TERM HOMEWORK

Subject: Recursion with MIPS Assembly

Due Date: 23rd May 2021, 23:59

Introduction

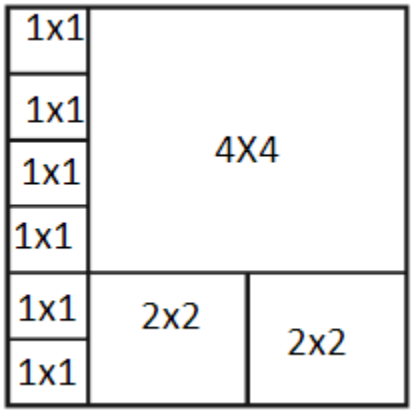
"In computer science, recursion is a method of solving a problem where the solution depends on solutions to smaller instances of the same problem. Such problems can generally be solved by iteration, but this needs to identify and index the smaller instances at programming time. Recursion solves such recursive problems by using functions that call themselves from within their own code." (Wikipedia)

In this homework, you will implement a MIPS Assembly equivalent of a recursive code about solving the problem "Minimum tiles of sizes in powers of two to cover whole area".

Problem

(Problem and definitions are obtained from: https://www.geeksforgeeks.org/minimum-tiles-of-sizes-in-powers-of-two-to-cover-whole-area/)

Given an area of N X M rectangle. You have the infinite number of tiles of size 2^i X 2^i, where i = 0, 1, 2,… . The objective is to find the minimum number of tiles to fill the given area with these smaller tiles.




```
Input : N = 5, M = 6.
Output : 9
Area of 5 X 6 can be covered with minimum 9 tiles.
6 tiles of 1 X 1, 2 tiles of 2 X 2, 1 tile of 4 X 4.
```

Python Code

The Python Code of the problem is given below (from geeksforgeeks.org). Please analyze the code and run it using a Python programming environment. You may double-check the results of your assembly code by using this Python code snippet. (use: https://www.programiz.com/python-programming/online-compiler/)

```
def minTiles(n, m):

    # base case, when area is 0.
    if n == 0 or m == 0:
        return 0

    # If n and m both are even, calculate
    # tiles for n/2 x m/2
    # Halfing both dimensions doesn't
    # change the number of tiles
    elif n%2 == 0 and m%2 == 0:
        return minTiles(int(n/2), int(m/2))

    # If n is even and m is odd
    # Use a row of 1x1 tiles
    elif n % 2 == 0 and m % 2 == 1:
        return (n + minTiles(int(n/2), int(m/2)))

    # If n is odd and m is even
    # Use a column of 1x1 tiles
    elif n % 2 == 1 and m % 2 == 0:
        return (m + minTiles(int(n/2), int(m/2)))

    # If n and m are odd add
    # row + column number of tiles
    else:
        return (n + m - 1 + minTiles(int(n/2), int(m/2)))
```

Driver Code (to check your function, main):

```
n = 5
m = 6
print (minTiles(n, m))
```

The output:

N=6, M=5 →	9
N=4, M=6 →	3
N=3, M=7 →	12
N=12, M=3 →	18

Environment and Document

You will use MARS Simulator as the development environment. You can download the simulator via the website (http://courses.missouristate.edu/KenVollmar/MARS/)

Restrictions:

- You should not use variables (defined in the data segment) inside of the recursive functions. Please use registers only. You may use it in the main function part if you think it is necessary.

Hints:

- Please check the Chapter 6 MIPS Assembly function call slides.
- Please check the recursive Fibonacci sample from "http://akomaenablog.blogspot.com/2008/04/mips-recursive-fibonacci.html" for a recursive function example.
- Another good example: https://people.cs.pitt.edu/~xujie/cs447/Mips/case_rec.html
- You may benefit from fib.asm (which is available inside of MARS simulator as an example, it is also shared via Webonline) for print functionality.

Important Notes:

- Your homework should be individual work, it is **not** a group project.
- Cheating or copy-paste from internet sources is strictly prohibited.
- Please be aware of the deadline.
- Your submission file structure is:
    - [StudentID] ->(it is a folder)
        - homework.asm

References:

- https://en.wikipedia.org/wiki/Recursion_(computer_science)
- https://www.geeksforgeeks.org/minimum-tiles-of-sizes-in-powers-of-two-to-cover-whole-area/