Echo

Write a MIPS assembly program to print the input string after saving it to the heap.

Input

A string containing ASCII characters

Output

The input string

Constraints

- The length of the string will not exceed 100 characters

Example

1) Hello world → Hello world

Requirements

- You must ensure that you save the input string to the heap

Marking Scheme

1M: Approach & Correctness (testcases will be given during demo)
1M: Code Cleanliness, use of appropriate comments

Binary Search

Write a MIPS assembly program to perform an iterative binary search to check if an element exists in a sorted array.

Input

n, [a1, a2, a3, …….an], x. Assume all values to be integers. The array is 0-indexed.

Output

"Yes at index __" if x is found in the array & "Not Found" if not

Constraints

- a1 <= a2 <= a3 <= a4 <= …… an
- 1 <= n <= 30

Example

1) 3 2 3 3 4 → Yes at index 1
2) 5 1 2 3 4 5 9 → Not found
3) 1 2 2 → Yes at index 0

Requirements

- The sorted array should be saved on the heap.
- The code should have O(logn) time complexity.
- Code must be iterative. Don't use recursion in this part of the assignment.

Marking Scheme

3M: Approach & Correctness (testcases will be given during demo)
1M: Code Cleanliness, use of appropriate comments

Fast Exponentiation

Write a MIPS assembly program to perform a recursive computation of x^n.

Input

x, n. Assume both to be 32-bit integers and that the result does not overflow.

Output

The result x^n

Constraints

- 0 <= x <= 10000
- 0 <= n <= 10000

Example

1) 2 4 → 16
2) 10 2 → 100

Requirements

- The code should have O(logn) time complexity
- The implementation must be recursive
- Function calls must follow caller-save and callee-save semantics

Marking Scheme

3M: Approach & Correctness (testcases will be given during demo)
1M: Code Cleanliness, use of appropriate comments

Submission Instructions

1. You need to create three files: “echo.asm”, “bin_search.asm”, and “fast_expo.asm”, for each of the programs, respectively. Zip them as <entry num 1>_<entry num 2>_A1.zip and submit it on Moodle. Only one teammate should submit.
2. Each of your programs should be able to take input and give the output when run on the qtspim simulator.
3. You can either extensively comment your code or prepare a write-up explaining your code and design decisions.