Computer Organization Project Report

1. Sel_Sort
    a. This function implements the selection sort algorithm to sort the array in ascending order.
    b. It uses nested loops to iterate over the array elements and compares adjacent elements to perform swapping if necessary.
    c. The outer loop (OuterLoop) controls the number of iterations, and the inner loop (InnerLoop) performs the actual comparison and swapping.
    d. The function takes two arguments: the base address of the array ($a0) and the length of the array ($a1).

Register	Funtionality
$a0	Refers to the starting memory address of the array being sorted.
$a1	Refers to the number of elements in the array.
$t0	Used for storing the base address of the array.
$t1	Used for storing the length of the array.
$t2	Used for the loop counter
$t3	Used for inner loop counter
$t4	Used for outer loop counter
$t5	Used for storing the result of array[j]<array[min_index]
$t6	Temporary register for storing element to be swapped.
$t7	Temporary register for storing element to be swapped.
$t8	Used for calculating memory addresses in inner loop.
$t9	Used for calculating memory addresses in outer loop.

2. Top_rank
    a. This function calls the Sel_Sort function and then returns the highest value in the sorted array.
    b. It first saves the return address and registers on the stack
    c. After calling Sel_Sort, it calculates the memory address of the last element in the sorted array using the base address and array length.
    d. The highest value is then loaded from memory and stored in register $v0.
    e. Finally, the saved registers and return address are restored, and the function returns.

Register	Functionality
$t0	Used for storing the base address of the array
$t1	Used for storing the address of the last element in the array

$v0	Used for returning the largest element in the array

3. Sem_abroad
   a. This function searches for the first value in the array that is less than 70 and returns its memory address.
   b. It saves the return address and registers on the stack.
   c. The function iterates over the array elements using a loop and compares each element to 70.
   d. If a value less than 70 is found, the memory address of that element is stored in register $v0.
   e. The saved registers and return address are then restored, and the function returns.

Register	FFuncnctioonalility
$t0	Used for storing the base address of the array.
$t1	Used for storing the length of the array.
$t2	Used for loop counter.
$t3	Used for comparison result.
$v0	Used for storing the base address of the new array (which starts from numbers bigger than or equal to 70)
$v1	Used for storing the number of the length of the new array