**COMP 3315 Lab4: Procedure Calls**

**Number and Name:**

1. **Object**: MIPS Procedure call conventions
2. **Procedure**:

Write and run a MIPS algorithm that inputs an array of integers at data segment arr1 and the code stores only the negative elements of arr1 to arr2. Use test procedure that inputs a single integer and returns 1 if integer is negative and 0 if it is positive by testing the MSB of the integer. Call test for each member of the array.

**Example**: *arr1*: .word 5, -1, 4, -6, 0, 8, -3, -7

arr2 : -1, -6, -3, -7

Write the C code first and then convert it to MIPS code and the output with short comments in the box below.

#include <stdio.h>

#define BITS sizeof(int) \* 8 // Total bits required to represent integer

int isValid(int num) {

int msb;

/\* Move first bit of 1 to highest order \*/

msb = 1 << (BITS - 1);

/\* Perform bitwise AND with msb and num \*/

if(num & msb){

//1

return 1;

}

else {

return 0;

}

}

int main()

{

int array1[8] = {5, -1, 4, -6, 0, 8, -3, -7};

int array2[4];

for (int i = 0; i < 8; i++) {

if(isValid(array1[i]) == 1) {

array2[i] = array1[i];

printf("%d", array2[i]);

}

}

return 0;

}