LAST NAME: *FIRST NAME :*

Take Home TEST 1 CSC 342 GRADED

Due date: April 5, 2021

Required for all students to submit

What to submit: Report, video < 2min on your report, and all source code files you have used.

1. Please hand write and sign statements affirming that you will not cheat here and in your submission:

"I will neither give nor receive unauthorized assistance on this TEST. I will use only one computing device to perform this TEST, I will not use cell while preforming this TEST".

2. **Students should allocate no more than total 12 hours to prepare report for this TEST.**

Please write
Start Time and date:
End TIME and date:

3. **You can use any resources.**
4. **You are not allowed to communicate with fellow students or other professionals.**

5. **THIS IS NOT A TEAM TEST!!**

You must submit:

1. **Report must have OBJECTIVE section, and Table of Content with links to corresponding sections.**
2. **Short Video presentation, no more than 2 min.**
3. **Source code and project files you have used. You must include README file with instructions on how to run your examples.**

The report file and video file names should be:
YOUR_LAST_NAME_TAKE_HOME_TEST_1,
LAST_NAME_VIDEO_TAKE_HOME_TEST_1

LAST NAME: *FIRST NAME :*

Take Home TEST 1 CSC 342 GRADED

Due date: April 5, 2021

Required for all students to submit

What to submit: Report, video < 2min on your report, and all source code files you have used.

Objective of the TEST

The objective of this TEST is to demonstrate student understanding and ability to compare MIPS instruction set architecture, Intel x86 ISA using Windows MS 32-bit compiler and debugger, and a Intel X86_64 bit ISA processor running Linux, 64 bit gcc and gdb,

In order to compare the instruction set architectures (ISA), students will be testing and analyzing programs described in Sections 2.1, 2.2, 2.3,2.4, 2.5, 2.6, 2.7, 2.8 of the textbook on all 3 platforms.

4. ***MIPS,*** *Student will be running the programs on the MARS MIPS simulator.*
5. ***Intel X32 ISA****, Windows 32-bit compiler, Student will be running programs on Visual Studio and using its debugger.*
6. ***X86_64 ISA, Linux 64-bit*** *platform, student will be running programs using 64 bit GCC and GDB debugger.*
   - **Explain little endian, and big endian where appropriate, demonstrate while loops , for loops, if-then-else statements on each platform, describe differences and similarities in each case.**
   - **For each example, students must display disassembly, registers, memory and explain.**
   - **Students must explain *LOCAL, Local STATIC*, STATIC variable addressing in each example case they present.**

Students may use examples shown below, or create their own examples.

Students MUST COVER SECTIONS 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8 from the textbook.

Students may use your own examples as long they cover all C programming structures. E.g. if then else, while loops, for loop, array handling with pointers and indexes, etc.

LAST NAME: *FIRST NAME :*

Take Home TEST 1 CSC 342 GRADED

Due date: April 5, 2021

Required for all students to submit

What to submit: Report, video < 2min on your report, and all source code files you have used.

Static local variables are a feature of C and other languages. The following example is must be included on all platforms used in comparison.

```c
/* static.c */
#include <stdio.h>
int natural_generator()
{
        int a = 1;
        static int b = -1;
        b += 1;
        return a + b;
}

int main()
{
        printf("%d\n", natural_generator());
        printf("%d\n", natural_generator());
        printf("%d\n", natural_generator());

        return 0;
}
```

LAST NAME: *FIRST NAME :*

Take Home TEST 1 CSC 342 GRADED

Due date: April 5, 2021

Required for all students to submit

What to submit: Report, video < 2min on your report, and all source code files you have used.

PART I

MIPS examples may be used

In order to run these programs on the MARS simulator, you have to write the programs all in MIPS assembly and then run the programs in MARS.

The first example is in Section 2.2 and is a simple program that is equivalent to running a = b+c and d = a-e in C.

2-2_1.asm

```
.data
a: .word 1
b: .word 2
c: .word 3
d: .word 4
e: .word 5
.text
lw $s0, a
lw $s1, b
lw $s2, c
lw $s3, d
lw $s4, e
# a = b + c
add $s0, $s1, $s2
sw $s0, a
# d = a - e
sub $s3, $s0, $s4
sw $s3, d
```

2-2_2.asm

```
.data
f: .word 0
g: .word 50
h: .word 40
i: .word 30
j: .word 20
.text
lw $s0, f
lw $s1, g
lw $s2, h
```

LAST NAME: *FIRST NAME :*

Take Home TEST 1 CSC 342 GRADED

Due date: April 5, 2021

Required for all students to submit

What to submit: Report, video < 2min on your report, and all source code files you have used.

```
lw $s3, i
lw $s4, j
# t0 = g+h

add $t0, $s1, $s2
# t1 = i+j
add $t1, $s3, $s4
# f = t0 - t1
sub $s0, $t0, $t1
sw $s0, f
```

2-3_1.asm

```
.data
g: .word 0
h: .word 22
A: .word 0-100
size: .word 100
.text
#just to set A[8] to 55
li $t1, 55
la $s3, A
sw $t1, 32($s3)
lw $s1, g
lw $s2, h
#loading the value of A[8] into t0
lw $t0, 32($s3)
add $s1, $s2, $t0
sw $s1, g
```

2-3_2.asm

```
.data
h: .word 25
A: .word 0-100
size: .word 100
.text
lw $s2, h
#initializing A[8] to 200
li $t1, 200
la $s3, A
sw $t1, 32($s3)
```

LAST NAME: *FIRST NAME :*

Take Home TEST 1 CSC 342 GRADED

Due date: April 5, 2021

Required for all students to submit

What to submit: Report, video < 2min on your report, and all source code files you have used.

```
#A[12] = h + A[8]
lw $t0, 32($s3)
add $t0, $s2, $t0
sw $t0, 48($s3)
```

2-5_2.asm

```
.data
h: .word 20
A: .word 0-400
size: .word 400
.text
la $t1, A
lw $s2, h
#initialzing A[300] to 13
li $t2, 13
sw $t2, 1200($t1)
lw $t0, 1200($t1) add
$t0, $s2, $t0
sw $t0, 1200($t1)
```

This code essentially performs the C equivalent of A[300] = h + A[300].

2.6

The next example is in section 2.6 and covers several logical operations that can be done in MIPS. You can use the code

2-6_1.asm

```
# left shift
li $s0, 9
sll $t2, $s0, 4
# AND
li $t2, 0xdc0
li $t1, 0x3c00
and $t0, $t1, $t2
# OR
or $t0, $t1, $t2
# NOR li
$t3, 0
nor $t0, $t1, $t3
```

LAST NAME: *FIRST NAME :*

Take Home TEST 1 CSC 342 GRADED

Due date: April 5, 2021

Required for all students to submit

What to submit: Report, video < 2min on your report, and all source code files you have used.

For section 2.8
Please write a simple "MAIN" in MIPS assembly that calls "myadd" function and analyze.

PART II.

Examples in x86 Intel on Windows 32-bit

In order to properly run these examples on a Windows 32-bit OS and debug them with Visual Studio's Debugger, students may write all these examples covered in MIPS in C. The first example is once again the first example in Section 2.2. Recall that this just does a = b+c and d = a-e. Here is the code You can write and use

2-2_1.c 1

```
void main(){
          static int a = 1;
          static int b = 2;
          static int c = 3;
          static int d = 4;
          static int e = 5;
          a = b + c;
          d = a - e;
          }
```

2-2_2.c 1

```
void main(){
          static int f = 0;
          static int g = 50;
          static int h = 40;
          static int i = 30;
          static int j = 20; 7
                  f = (g + h) - (i
          + j);
  }
```

LAST NAME: *FIRST NAME :*

Take Home TEST 1 CSC 342 GRADED

Due date: April 5, 2021

Required for all students to submit

What to submit: Report, video < 2min on your report, and all source code files you have used.

2-3_1.c 1

```
void main(){
          static int g = 0;
          static int h = 22;
          static int A[100];
          A[8] = 55;
          g = h + A[8];
          }
```

2-3_2.c 1

```
void main(){
          static int h = 25;
          static int A[100];
          A[8] = 200;
          A[12] = h + A[8];
          }
```

2-5_1.c 1

```
void main(){
          static int a = 0;
          static int b = 0;
          static int c = 0;
          a = b + c;
          }
```

2-6_1.c 1

```
void main(){
          static int s0 = 9;
          static int t1 = 0x3c00;
          static int t2 = 0xdc0;
          static int t3 = 0;
t3 = s0 << 4;
static int t0 = 0;
t0 = t1 & t2;
t0 = t1 | t2;
t0 = ~t1;
}
```

LAST NAME: *FIRST NAME :*

Take Home TEST 1 CSC 342

Due date: April 5, 2021

What to submit: Report, video < 2min on your report, and all source code files you have used.

2-6_1.asm

```
    # left shift
    li $s0, 9
    sll $t2, $s0, 4
    # AND
    li $t2, 0xdc0
    li $t1, 0x3c00
and $t0, $t1, $t2
# OR
or $t0, $t1, $t2
# NOR li
$t3, 0
nor $t0, $t1, $t3
```

6. The following program must be analyzed:

```
/* static.c */
#include <stdio.h>
int natural_generator()
{
        int a = 1;
        static int b = -1;
        b += 1;
        return a + b;
}

int main()
{
        printf("%d\n", natural_generator());
        printf("%d\n", natural_generator());
        printf("%d\n", natural_generator());

        return 0;
}
```

2.8 Write and analyze in Debug mode example for “MAIN” calls simple “myadd” function.

PART III.

Repeat all the above PART I, and PART II for LINUX, gcc, gdb 64 bit. On Intel X86-64 ISA.