Project 2 Report

Christopher Badolato

badolato.christopher@knights.ucf.edu

EEL 3801-0011: Computer Organization

3/18/2019

1.0 Project Description

For this report we will be designing a histogram from a string of characters using only the letters "u, c, f, k, n, i, g, h, t, s". We will loop through our string comparing each character to the ascii value of the characters we are searching for. Our only functional requirement is, the string hardcoded into the program must be entirely lowercase for the program to execute correctly. After we have our frequency for each letter, we will then create a histogram for each letter using hashtags (#). Next, we can calculate our CPI, total instructions, and Energy consumption for our MIPS program. Finally, we will create a LESS efficient version of the MIPS program and compared our total energy consumptions.

2.0 Program Design

To begin, we much create our string variable within our .data as well create string variables for our letters "u,c,f,k,n,i,g,h,t,s" and our "#" for printing later. In our case we will be hardcoding a string into the program and naming it "str". The hardcoded string str must be entirely lowercase. We will then load our variable "str" address into our register $a0 as well as initialize registers $s0, and $t0-$t9 to zero.

Our loopThoughString loop now begins. We will first, load the first element our string 0($a) into our $s0 register. Then we will introduce our exit condition of our loopThroughString which is, if our value stored in $s0 is equal to zero then we have reached the end of our character string. Otherwise we will proceed to calculate character frequency.

We will now essentially incorporate "if-else" statements. Comparing our $s0 value, (current character ascii value) against the ascii value of the character we are trying to find the frequency of (we can follow the table below for the ascii values of the characters we need). If our values are NOT EQUAL, we will immediately branch to the next character we would like to find the frequency of. If our values are EQUAL, we will add to the register counter for that character. We can follow the symbol table below to see in which register we will store the frequency for each character. This will continue for each character until we reach nextChar10: which will add one to our $a0 register incrementing our string, then will jump us back to the start of loopThroughString. We can follow the flowchart Figure 1 below to get a better understanding of how the program runs.

After the loop has finished creating a count for each letter, we can print the frequency of each character using $v0 to call the type we would like to print. We then can move our integer or load the address of our string values to print into our $a0 overwriting the register since we now have our frequency count and no longer need the string.

Finally, we need to print our hashtag histogram. Doing so requires us to loop through the frequency count of each character to print the exact number of hashtags, this will create a simple graph. We have our frequency for each character so, we only need a register to store the iteration of each loop ($s1). For each character we print a hashtag for each iteration of the loop. We will add one to our iterator, once reaches the current characters frequency value the loop will branch to the next character. For each character we intend to make a histogram bar for we must reset this iterator ($s1) to zero to reset our loop count to zero. Using our printLoop1 and nextLoop2 we will create loops using a branch that will check if our iterator is equal to the frequency of the current character. We can follow the flowchart in Figure 2 for a better understand of how each print works.

As for the unoptimized version of the code I created a version in which the program would not just iterate through the string once to take the frequency, but twice. The program would loop through the string to take a count of ONLY the character ‘u’. That way we are forced to run through the string AGAIN to take the frequency count for the rest of the characters (c,f,k,n,i,h,g,t,s). The total dynamic instruction count will therefore increase, increasing the total energy consumption of the program.

Figure 1

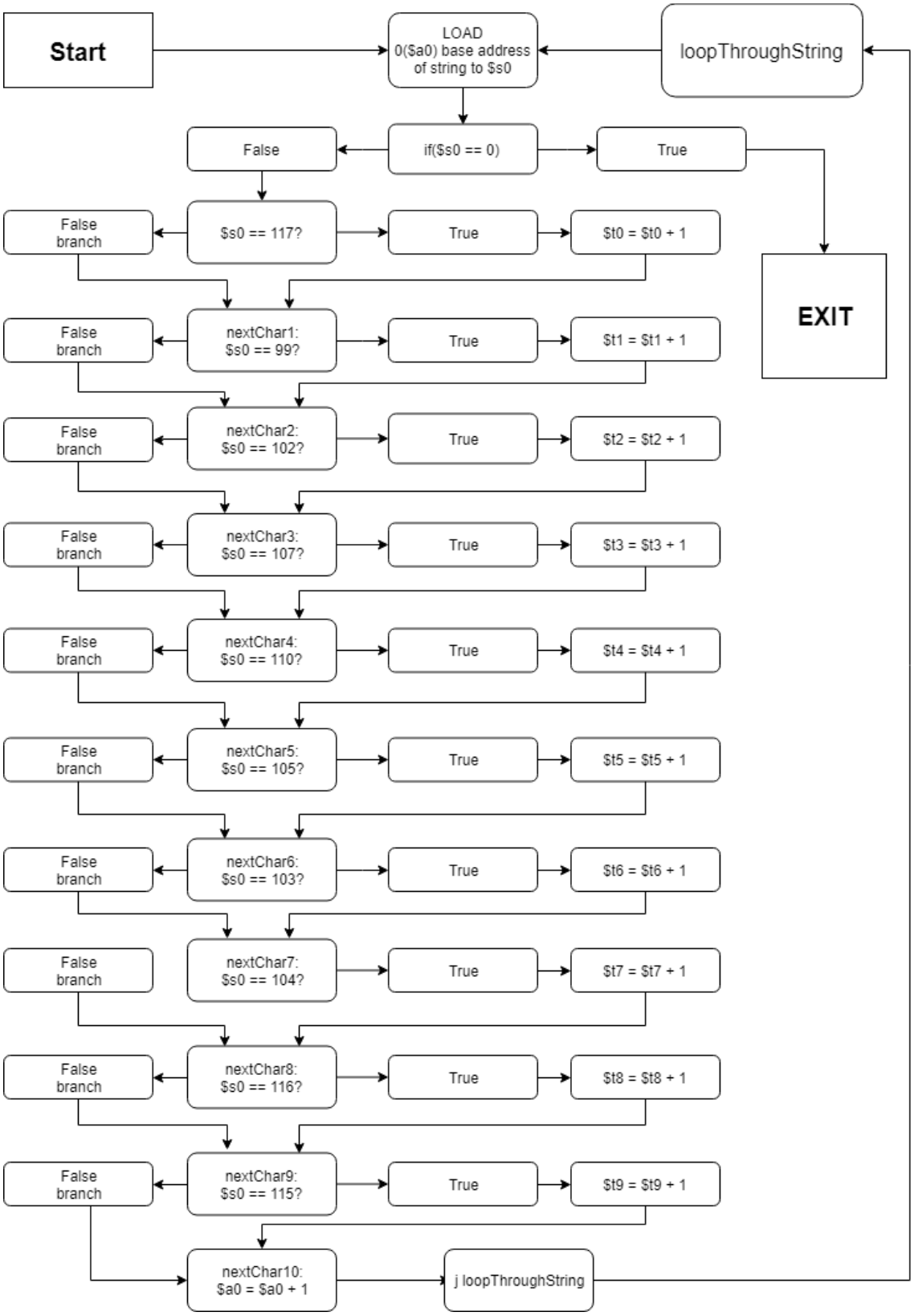
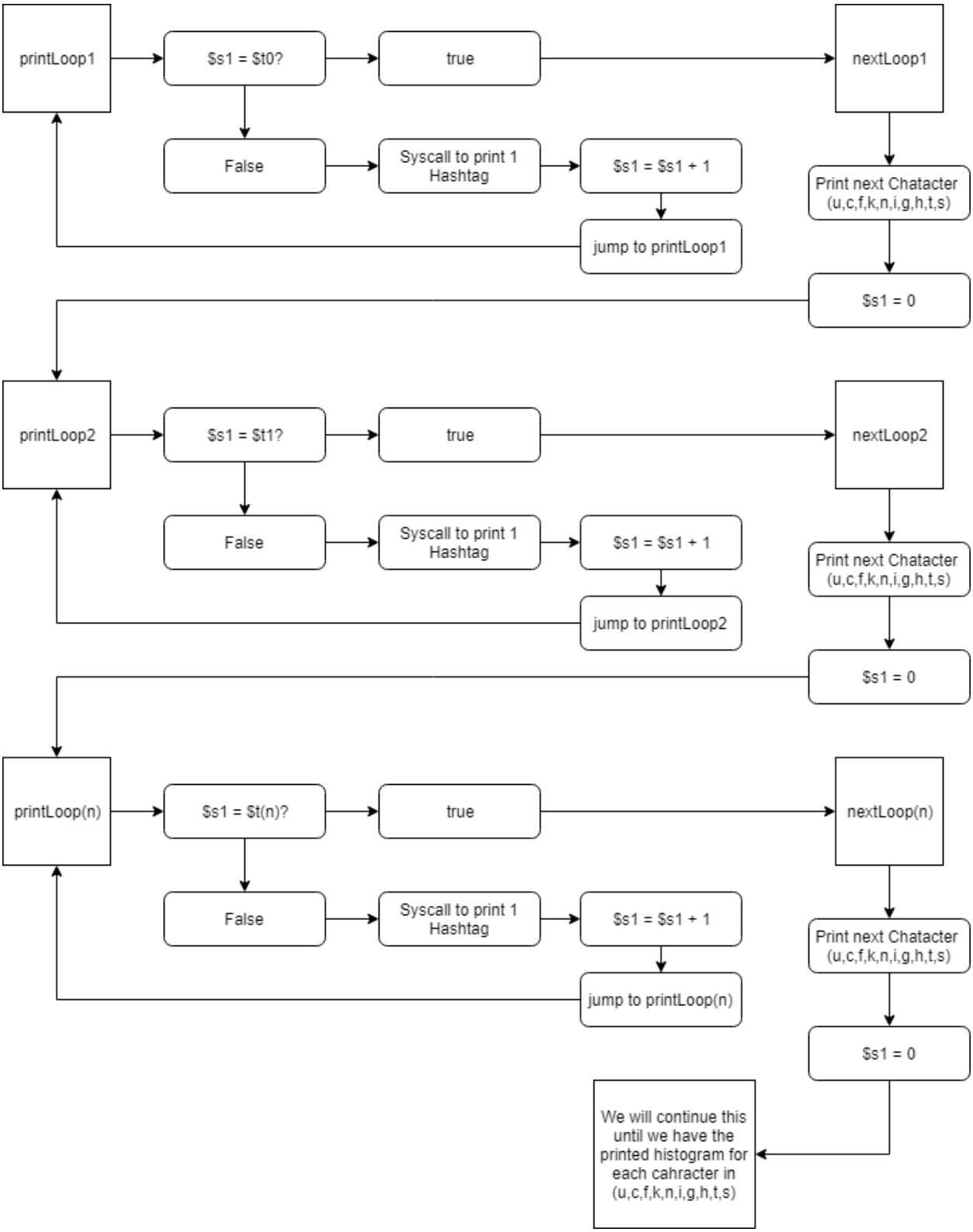

Start
LOAD
0($a0) base address
of string to $s0
loopThroughString
if($s0 == 0)
False
True
EXIT
$s0 == 117?
False branch
True
$t0 = $t0 + 1
nextChar1:
$s0 == 99?
$t1 = $t1 + 1
nextChar2:
$s0 == 102?
$t2 = $t2 + 1
nextChar3:
$s0 == 107?
$t3 = $t3 + 1
nextChar4:
$s0 == 110?
$t4 = $t4 + 1
nextChar5:
$s0 == 105?
$t5 = $t5 + 1
nextChar6:
$s0 == 103?
$t6 = $t6 + 1
nextChar7:
$s0 == 104?
$t7 = $t7 + 1
nextChar8:
$s0 == 116?
$t8 = $t8 + 1
nextChar9:
$s0 == 115?
$t9 = $t9 + 1
nextChar10:
$a0 = $a0 + 1
j loopThroughString

Figure 2


printLoop1
$s1 = $t0?
true
nextLoop1
False
Syscall to print 1 Hashtag
$s1 = $s1 + 1
Print next Chatacter (u,c,f,k,n,i,g,h,t,s)
jump to printLoop1
$s1 = 0
printLoop2
$s1 = $t1?
true
nextLoop2
False
Syscall to print 1 Hashtag
$s1 = $s1 + 1
Print next Chatacter (u,c,f,k,n,i,g,h,t,s)
jump to printLoop2
$s1 = 0
printLoop(n)
$s1 = $t(n)?
true
nextLoop(n)
False
Syscall to print 1 Hashtag
$s1 = $s1 + 1
Print next Chatacter (u,c,f,k,n,i,g,h,t,s)
jump to printLoop(n)
$s1 = 0
We will continue this until we have the printed histogram for each cahracter in (u,c,f,k,n,i,g,h,t,s)

3.0 Symbol and Label Tables

Symbol Table

Register	Assignment
$v0	Tells Mars when to Syscall.
Sa0	Base address of our strings used throughout the program add $a0, $a0, 1 takes us to the next of our string before we loop Also stores our strings to print later in the program. (Used as temporary string storage)
$s0	Stores the value of each string character as we loop through our string.
$t0-$t9	Stores the integer count of each character "u, c, f, k, n, i, g, h, t, s" respectively.
$s1	Used as our repeated loop counter throughout the printing of the hashtag histogram.

Character	register	Ascii Value
u	$t0	117
c	$t1	99
f	$t2	102
k	$t3	107
n	$t4	110
i	$t5	105
g	$t6	103
h	$t7	104
t	$t8	116
s	$t9	115

Label Table

Label	Use
loopThroughString: exit:	Represents a while loop, which loops through our string. (although in the c code prototype I used a for loop and string length). We add to our base address of the string to continue this loop to the next character. We will exit this loop only if our value stored within our loaded bit register is null.
nextChar1: - nextChar10:	Within the above loop we essentially create if else statements If is equal, add 1 to our count at that character, check the next character.

printLoop1: - printLoop10	JUMPS us to the top of our current print loop so that we print the correct number of hashtags(#) in our histogram.
nextLoop1: - nextLoop10	Branches to our next characters hashtag printing loop once our loop counter reaches the total character count.

4.0 Project 2: part B

Sentence 1:
the knights field 16 varsity teams (6 men's, 9 women's and one co-ed sport) that have won numerous national and conference titles. the ucf knights football team won conference championships in 2007 and 2010, and the knights women's basketball team won conference titles the 2009 and 2010.

DIC: 8016
r-Type: 171
I type: 7448
J-type: 397
CPI calculation sentence 1:

$$\frac{(171\ x\ 4)\ +\ (7448\ x\ 2)\ +\ (397)}{8016}$$

CPI = 1.99

Energy Consumption:
ALU: 3871
Jump: 397
Branch: 3288
Memory: 289
Other: 171
Energy Consumption Calculation sentence 1:
(3871 x 3fj) + (397 x 2fj) + (3288 x 5fj) + (289 x 100fj) + (171 x 6fj) =
Energy consumption = 58773 fj

Sentence 2:

barring any last-minute snags, the jacksonville jaguars are expected to sign free-agent quarterback nick foles to a multiyear contract when the new league year opens next week, according to sources.

DIC: 5592
r-Type: 138
I type: 5180
J-type: 274
CPI calculation sentence 2:

$$\frac{(138\ x\ 4)\ +\ (5180\ x\ 2)\ +\ (274)}{5592}$$

CPI = 2.0004

Energy Consumption:
ALU: 2716
Jump: 274
Branch: 2265
Memory: 199
Other: 138
Energy Consumption Calculation sentence 2:
(2716 x 3fj) + (274 x 2fj) + (2265 x 5fj) + (199 x 100fj) + (139 x 6fj) =
Energy consumption = 40755 fj

Sentence 3:
at the beginning of the process, i told this guy i didn't want [an opt-out] anywhere i went," harper said, motioning toward boras. "i wanted to be able to dig my roots somewhere, that was through the good, through the bad, through the ups and downs of the organization. it's going to be tough for 13 years to win every single year, and i totally understand that.

DIC: 9944
r-Type: 196
I type: 9254
J-type: 397
CPI calculation sentence 3:

$$\frac{(196\ x\ 4)\ +\ (9254\ x\ 2)\ +\ (397)}{9944}$$

CPI = 1.98

Energy Consumption:
ALU: 4788
Jump: 494
Branch: 4105
Memory: 361
Other: 196
Energy Consumption Calculation sentence 3:
(4788 x 3fj) + (494 x 2fj) + (4105 x 5fj) + (361 x 100fj) + (196 x 6fj) =
Energy consumption = 73153 fj

Sentence 4:

that other guy would be angels outfielder mike trout, who many believe could earn a contract that dwarfs harper's in total value.
DIC: 3672
r-Type: 105
I type: 3395
J-type: 172
CPI calculation sentence 4:

$$\frac{(105\ x\ 4)\ +\ (3395\ x\ 2)\ +\ (172)}{3672}$$

CPI =2.01

Energy Consumption:
ALU: 1792
Jump: 172
Branch: 1473
Memory: 130
Other: 105
Energy Consumption Calculation sentence 4:
(1792 x 3fj) + (172 x 2fj) + (1473 x 5fj) + (130 x 100fj) + (105 x 6fj) =
Energy consumption = 26715 fj

sentence 5:
looming just beyond the outfield and over harper's left shoulder was his mean-mugging face, edited into an image on display on the spectrum field video board. on the protective over-dugout netting just behind harper and others sitting at a red-clothed table on the dugout -- boras, phillies owner john middleton and klentak -- were posters of magazines depicting harper from cover shoots he had with espn the magazine and sports illustrated.
DIC: 12096
r-Type: 222
I type: 11273
J-type: 601

CPI calculation sentence 5:

$$\frac{(222\ x\ 4)\ +\ (11273\ x\ 2)\ +\ (601)}{12096}$$

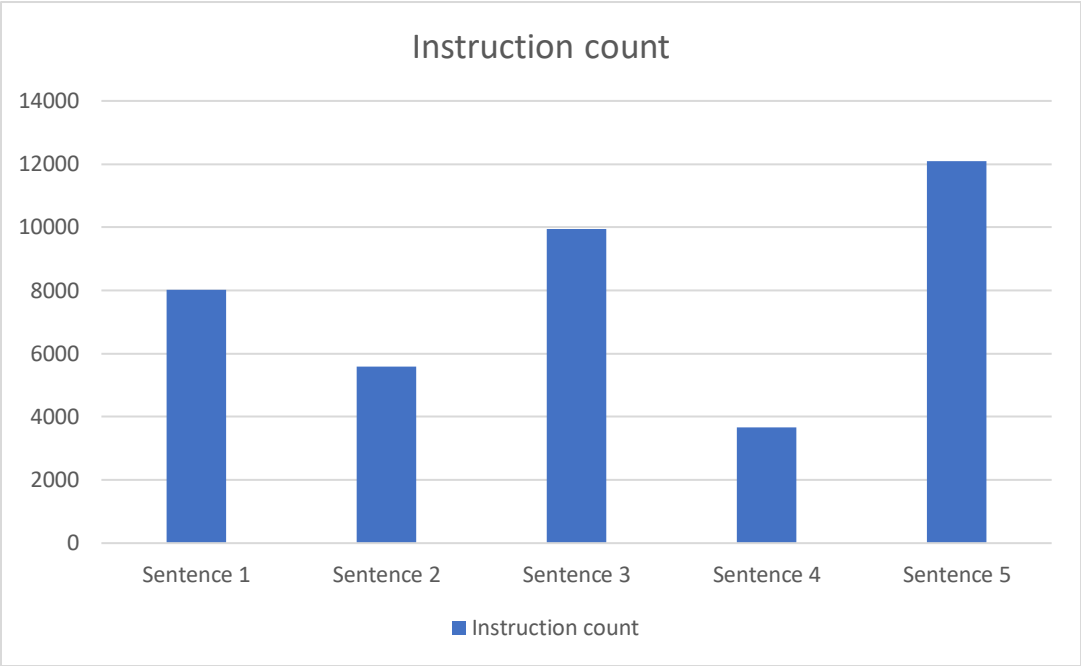
CPI = 1.987

Energy Consumption:
ALU: 5809
Jump: 601
Branch: 5022
Memory: 442
Other: 222

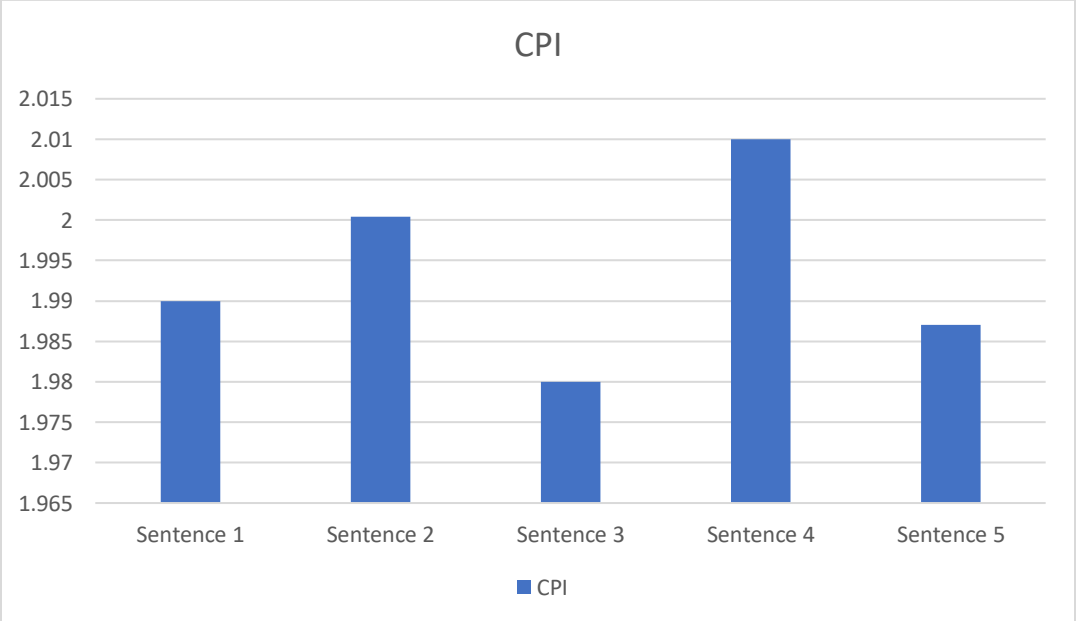
Energy Consumption Calculation sentence 5:
(5809 x 3fj) + (601 x 2fj) + (5022 x 5fj) + (442 x 100fj) + (222 x 6fj) =
Energy consumption = 89271 fj

Below we have a graphical representation for each sentence. All these sentences were used as test cases throughout the project report.

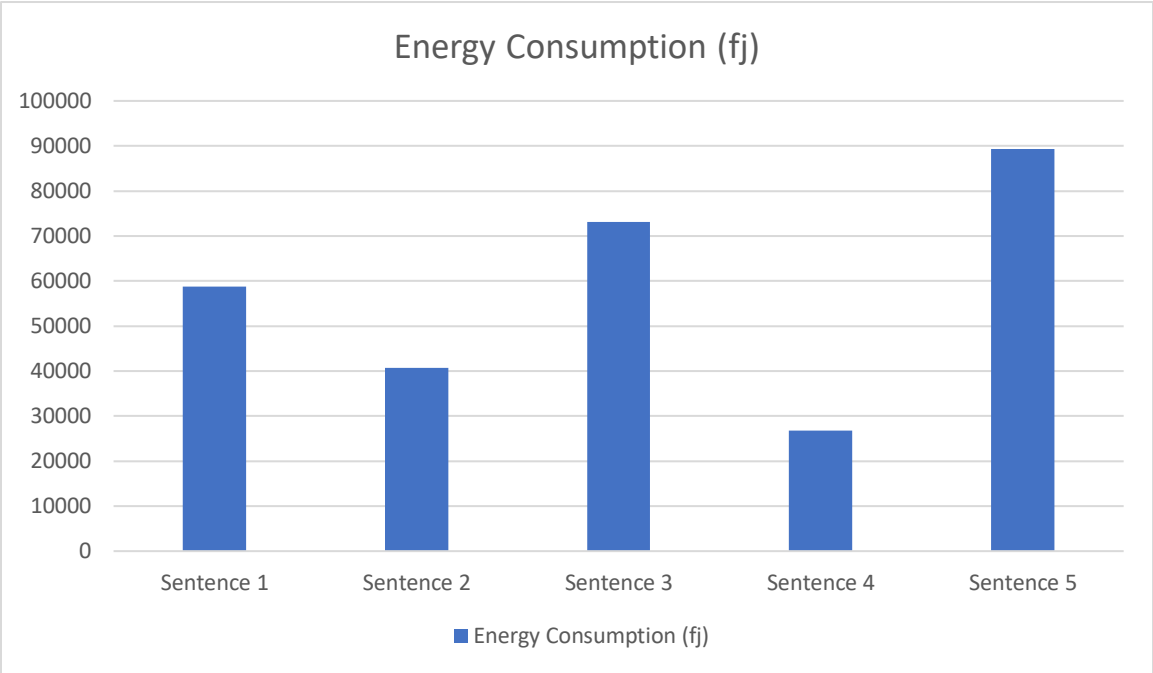
Graph of our instruction count totals of our MIPS program



Clocks per instruction of the program for each sentence



Total Energy consumption of the original program



Energy consumption of unoptimized code

To create some difference in our instruction count I created a second program that would, loop through the string ONLY for the letter u. Going through the string once for u, then proceeding to fill the other histogram values with a second pass through of our string. Therefore, increasing the number of times we run through the string, increasing the instruction count, and increasing the total energy consumption.

Below are the calculation of energy for the Altered program

Energy Consumption Calculation sentence 1:

(4158 x 3fj) + (682 x 2fj) + (3571 x 5fj) + (575 x 100fj) + (171 x 6fj) =

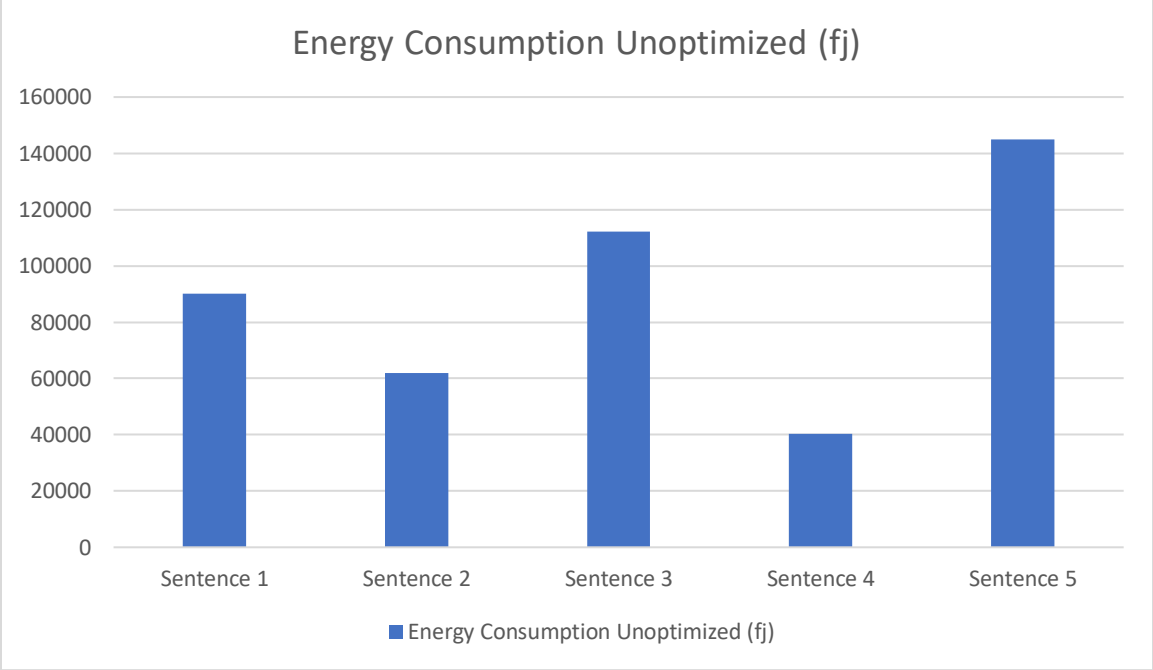
Energy consumption = 90219 fj

Energy Consumption Calculation sentence 2:

(2910 x 3fj) + (446 x 2fj) + (2452 x 5fj) + (392 x 100fj) + (138 x 6fj) =

Energy consumption =61910 fj

Energy Consumption Calculation sentence 3: (5154 x 3fj) + (848 x 2fj) + (4462 x 5fj) + (716 x 100fj) + (196 x 6fj) = Energy consumption = 112244 fj
Energy Consumption Calculation sentence 4: (1917 x 3fj) + (295 x 2fj) + (1591 x 5fj) + (254 x 100fj) + (105 x 6fj) = Energy consumption = 40326 fj
Energy Consumption Calculation sentence 5: (6241 x 3fj) + (1031 x 2fj) + (5442 x 5fj) + (873 x 100fj) + (222 x 6fj) = Energy consumption = 144877 fj



5.0 Learning Coverage

- Looping through a string in MIPS from a base address to store the current character into a register space for comparisons.
- We can check if our character is null or 0 to find the end of our string.
- We've learned the ability to compare characters of a string to their ascii value to separate out certain characters or in our case, take a count.

- Utilizing loops to create and print a histogram of any ascii character in MIPS.
- A little thing such as looping through a string an extra time can have a larger effect on the program, hardware, and energy consumption of your system.

6.0 Prototype in C

Code can be easily copy and pasted into IDE.

```
// Christopher Badolato 3064088
// 3/18/2019
// EEL3801 0011
/* this code will conduct the given functions from project 2.
It represents the MIPS code we are writing, which involves creating a histogram
using only the letters from "UCF KNIGHTS" the string entered must be LOWERCASE
*/
#include <string.h>
#include <stdio.h>

int main(){
      //created integer variables to represent the count at each letters
      //as well as grabs our string length, This code only works if entered string
      //is all lowercase
   int loopCounter = 0, strLen, charAsciiValue;
   int u = 0, c = 0, f = 0, k = 0, n = 0 ,i = 0 ,g = 0, h = 0, t = 0,s = 0;
   char* string = "the knights field 16 varsity teams (6 men's, 9 women's and one co-ed sport) that have
won numerous national and conference titles. the ucf knights football team won conference
championships in 2007 and 2010, and the knights women's basketball team won conference titles the
2009 and 2010.";
   strLen = strlen(string);
      //Checks each char in the string  to see if it is part of our UCFKNIGHTS char list
      // we will add 1 to the count if the "see" one of our selected characters.
   for(loopCounter = 0; loopCounter < strLen; loopCounter++){
      charAsciiValue = (int)string[loopCounter];
         //u
      if(charAsciiValue == 117){
         u++;
      }
         //c
      else if(charAsciiValue == 99){
         c++;
      }
         //f
      else if(charAsciiValue == 102){
         f++;
      }
         //k
      else if(charAsciiValue == 107){
         k++;
      }
```

```
        //n
    else if(charAsciiValue == 110){
        n++;
    }
        //i
    else if(charAsciiValue == 105){
        i++;
    }
        //g
    else if(charAsciiValue == 103){
        g++;
    }
        //h
    else if(charAsciiValue == 104){
        h++;
    }
        //t
    else if(charAsciiValue == 116){
        t++;
    }
        //s
    else if(charAsciiValue == 115){
        s++;
    }
        //Go to the next character in the string once we have checked
        //each character we'd like to make the histogram for. "ucfknights"
    else{
        continue;
    }
}
    //prints our integer histogram.
printf("u: %d\n", u);
printf("c: %d\n", c);
printf("f: %d\n", f);
printf("k: %d\n", k);
printf("n: %d\n", n);
printf("i: %d\n", i);
printf("g: %d\n", g);
printf("h: %d\n", h);
printf("t: %d\n", t);
printf("s: %d\n", s);
printf("\n");

    //Based on the values from the histogram above we will print pounds or
    //"hashtags" for each character count by looping through that character count.
    //u
printf("u: ");
for(loopCounter = 0; loopCounter < u; loopCounter++){
    printf("#");
}
printf("\n");
```

```
    //c
printf("c: ");
for(loopCounter = 0; loopCounter < c; loopCounter++){
    printf("#");
}
printf("\n");
    //f
printf("f: ");
for(loopCounter = 0; loopCounter < f; loopCounter++){
    printf("#");
}
printf("\n");
    //k
printf("k: ");
for(loopCounter = 0; loopCounter < k; loopCounter++){
    printf("#");
}
printf("\n");
    //n
printf("n: ");
for(loopCounter = 0; loopCounter < n; loopCounter++){
    printf("#");
}
printf("\n");
    //i
printf("i: ");
for(loopCounter = 0; loopCounter < i; loopCounter++){
    printf("#");
}
printf("\n");
    //g
printf("g: ");
for(loopCounter = 0; loopCounter < g; loopCounter++){
    printf("#");
}
printf("\n");
    //h
printf("h: ");
for(loopCounter = 0; loopCounter < h; loopCounter++){
    printf("#");
}
printf("\n");
    //t
printf("t: ");
for(loopCounter = 0; loopCounter < t; loopCounter++){
    printf("#");
}
printf("\n");
    //s
printf("s: ");
for(loopCounter = 0; loopCounter < s; loopCounter++){
```

```
        printf("#");
    }
    return 0;
}
```

7.0 Test Plan

To test our Mips code, we have applied various test cases to ensure our output is correct. Each case corresponds to the results shown below in the images. We have tested and verified with our C code that these outputs correspond to the output of our MIPS run version. We hardcoded the strings directly into the program.

Inputs (string)	Outputs (histogram)
the knights field 16 varsity teams (6 men's, 9 women's and one co-ed sport) that have won numerous national and conference titles. the ucf knights football team won conference championships in 2007 and 2010, and the knights women's basketball team won conference titles the 2009 and 2010.	u: 3 c: 9 f: 6 k: 4 n: 26 i: 11 g: 3 h: 11 t: 21 s: 15 u: ### c: ######### f: ###### k: #### n: ########################## i: ########### g: ### h: ########### t: #################### s: ###############
barring any last-minute snags, the jacksonville jaguars are expected to sign free-agent quarterback nick foles to a multiyear contract when the new league year opens next week, according to sources.	u: 6 c: 9 f: 2 k: 4 n: 14 i: 7 g: 7 h: 3 t: 14 s: 10 u: ###### c: ######### f: ## k: ####

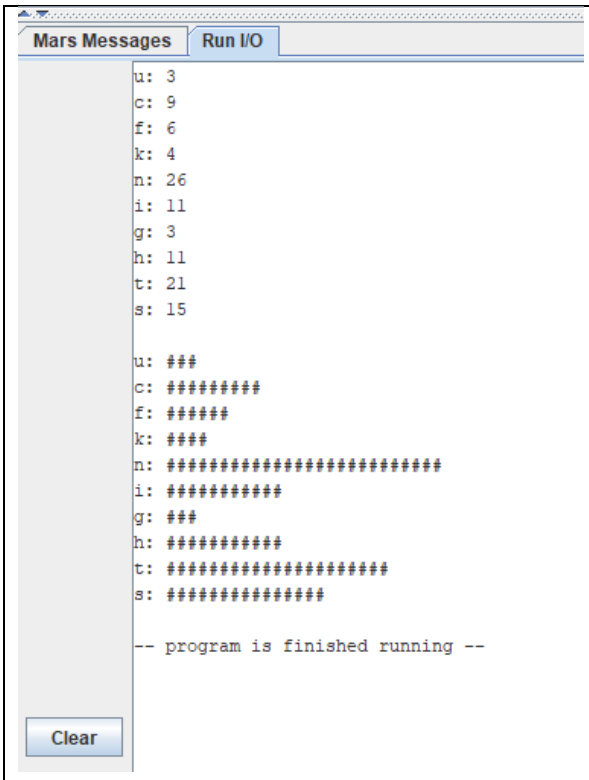
	n: ############## i: ####### g: ####### h: ### t: ############## s: ##########
at the beginning of the process, i told this guy i didn't want [an opt-out] anywhere i went, harper said, motioning toward boras. i wanted to be able to dig my roots somewhere, that was through the good, through the bad, through the ups and downs of the organization. it's going to be tough for 13 years to win every single year, and i totally understand that.	u: 8 c: 1 f: 3 k: 0 n: 21 i: 19 g: 14 h: 19 t: 35 s: 14 u: ######## c: # f: ### k: n: #################### i: ################## g: ############## h: ################## t: #################################### s: ##############
that other guy would be angels outfielder mike trout, who many believe could earn a contract that dwarfs harper's in total value.	u: 6 c: 3 f: 2 k: 1 n: 5 i: 4 g: 2 h: 5 t: 12 s: 3 u: ###### c: ### f: ## k: # n: ##### i: #### g: ## h: ##### t: ############ s: ###
looming just beyond the outfield and over harper's left shoulder was his mean-mugging face, edited	u: 11 c: 6

into an image on display on the spectrum field video board. on the protective over-dugout netting just behind harper and others sitting at a red-clothed table on the dugout -- boras, phillies owner john middleton and klentak -- were posters of magazines depicting harper from cover shoots he had with espn the magazine and sports illustrated.	f: 6 k: 2 n: 26 i: 24 g: 12 h: 19 t: 33 s: 21 u: ########## c: ###### f: ###### k: ## n: ########################## i: ######################## g: ############ h: ################## t: ################################ s: ######################

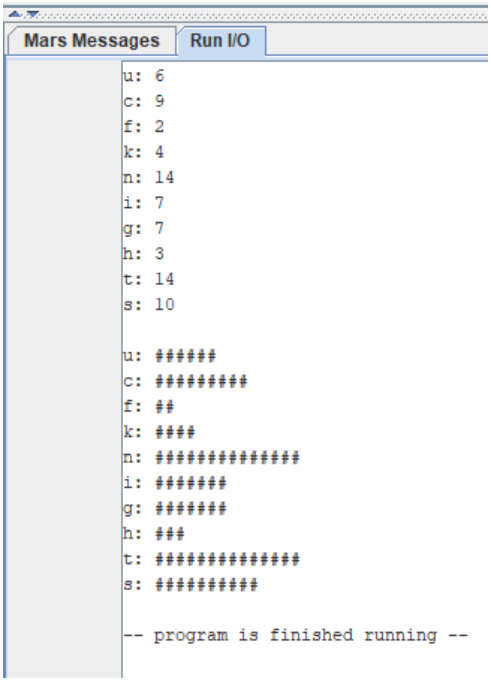
8.0 Test Results

Following the input strings from above we receive these outputs below. We can see that each character is printed above with the frequency of each character. As well as our hashtag histogram.

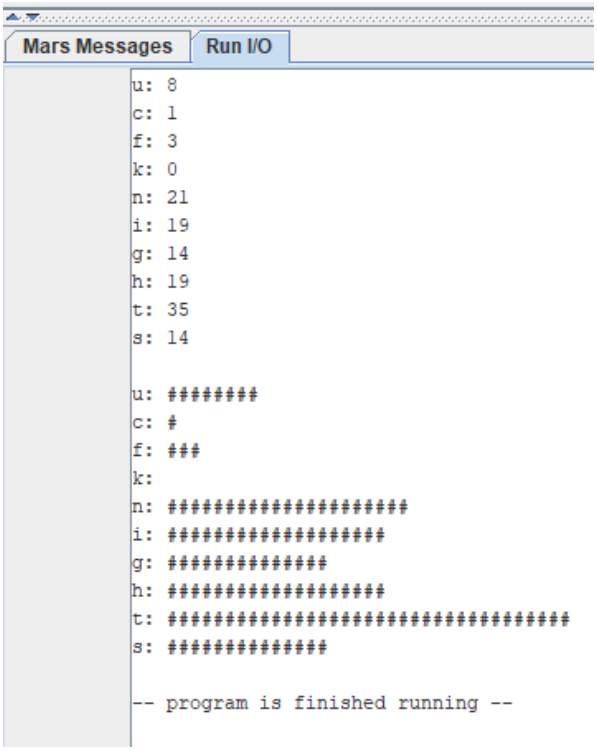
Testcase 1 (Sentence 1)



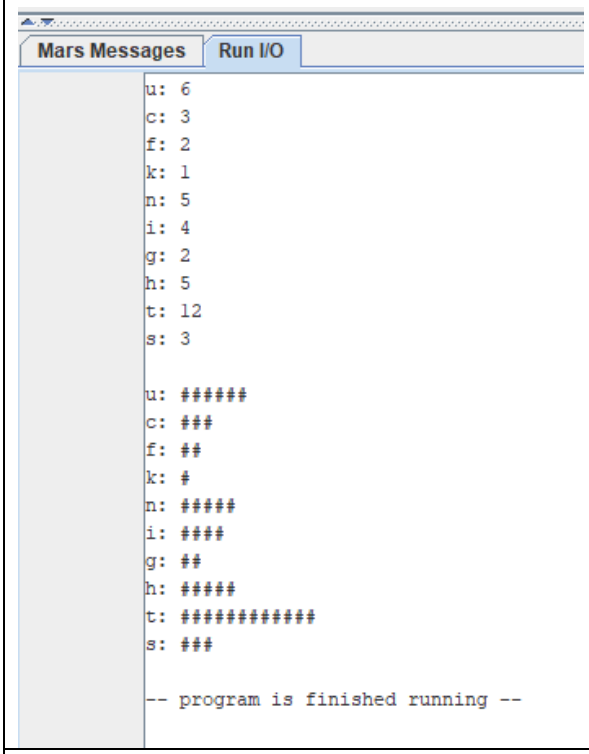
Testcase 2 (Sentence 2)



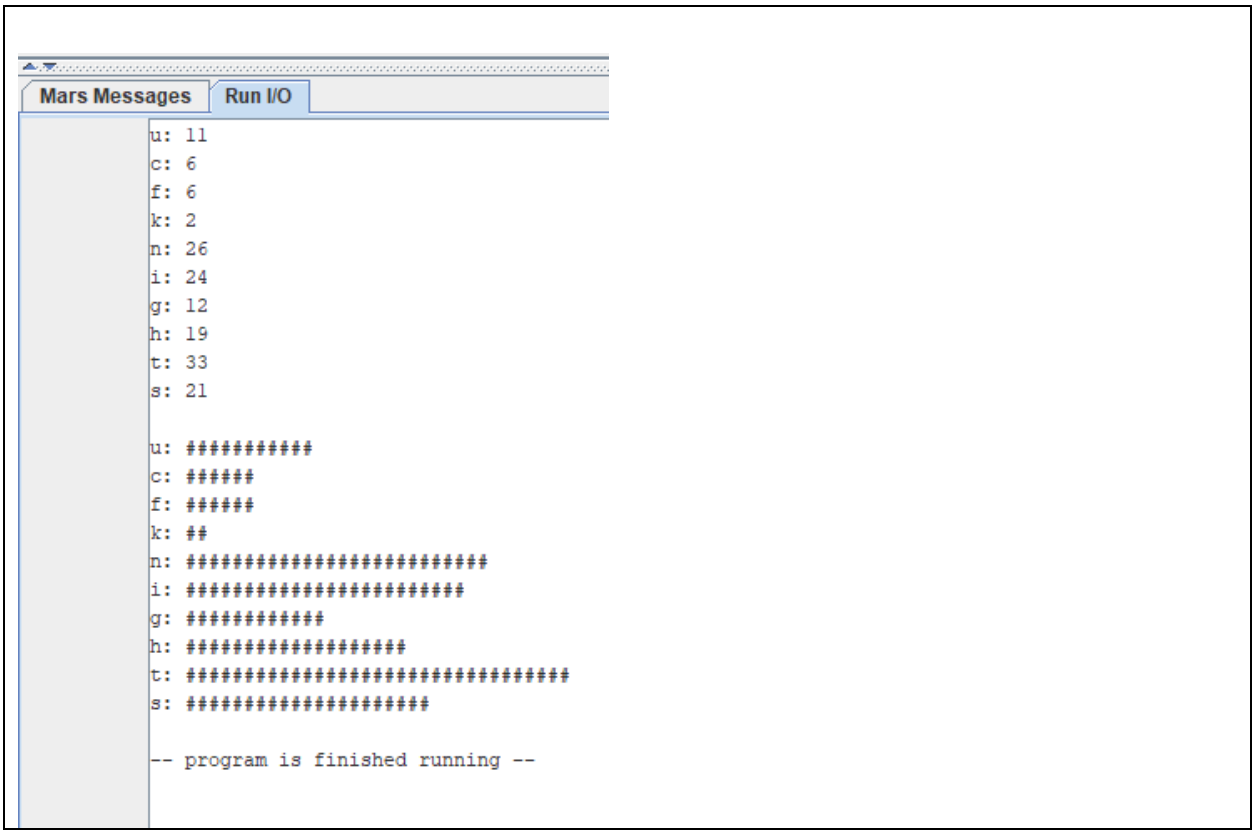
Testcase 3 (Sentence 3)



Testcase 4 (Sentence 4)



Testcase 5 (Sentence 5)



9.0 References

- EEL 3801 00219 DeMara Slides, Module 03-MIPS-ISA.pdf.
- Mars 4.5 Mips Simulator
- Recitation files
- Testing reference Sheet