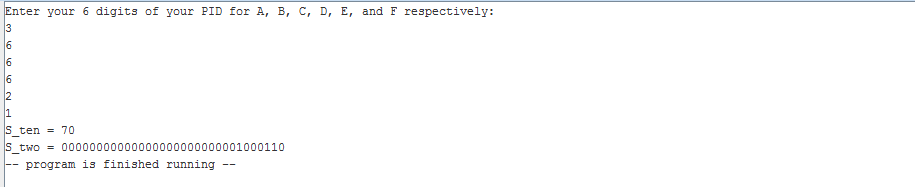
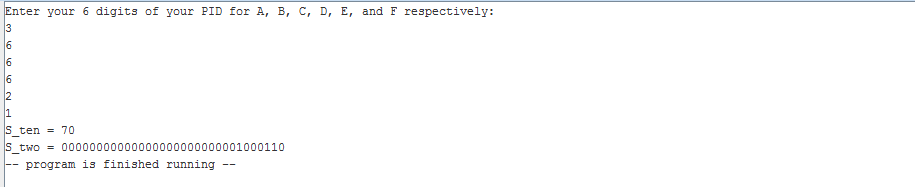
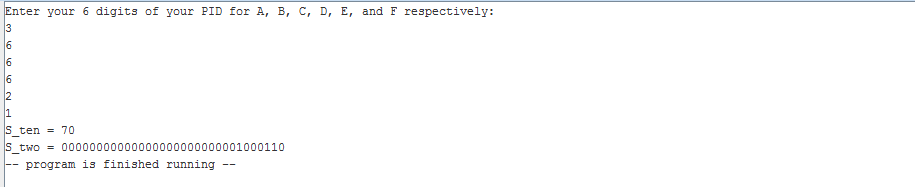
Lab Report 2

1. Project Description:
   1. This project is used to learn basic math computation in assembly language for the MIPS processor. In this lab the student will use registers to store data values for ints and strings for output and computation. Learn how to use loops to use the mathematical multiplication operation using addition. The goal is to output a number in base ten and base two after doing some mathematical computation to inputs.
2. Program Design:
   1. Save data to variables. Output the first string asking for user input. Ask for user input and store in registers. Complete the mathematical computations on the data to find S. Output string saying the variable s in base two is, the output the int s in base ten. Output the string for s in base two then output the number s in base two.
   2. Save data to variables -> Output the first string asking for user input -> Ask for user input and store in registers -> Complete the mathematical computations on the data to find S -> Output string saying the variable s in base two is, the output the int s in base ten -> Output the string for s in base two then output the number s in base two.
3. Symbol table:

|  |  |
| --- | --- |
| $v0 | This is used for the system commands such as 4, output string. |
| $t0-$t7 | These are used as data registers. This could be used to store a number after input or mathematical computation. |
| $a0 | Use this register for when outputting a string. Load the string on this register to output it. |

|  |  |
| --- | --- |
| Exit(1-4): | Used to say to exit a loop when the condition is met. |
| J L(1-4) | Telling the code to loop back to top of the loop. |

1. Learning Coverage:
   1. How to use loop to preform multiplication in assembly language.
   2. How to convert loops in C to assembly.
   3. How to output ints in different bases, specifically base two and ten.
   4. How to take in user input in assembly.
   5. The difference between writing for and while in assembly.
2. Prototype in C-language:
   1. #include <stdio.h>
   2. Main() {
      1. Int S, A, B, C, D, E, F, x,y;
      2. printf(“Enter your 6 digits of your PID for A, B, C, D, E, and F respectively:\n”);
      3. scanf(“%d, %d, %d, %d, %d, %d”, &A, &B, &C, &D, &E, &F);
      4. S = (A x B) + (18 x C) – (9 x D) – (E x F);
      5. printf(“S\_ten = %d”, S);
      6. printf(“S\_two = );
      7. for (x=31; x>=0;x--) {
         1. y = s >>x;
         2. if(k &1)
            1. printf(“1”);
         3. else {
            1. printf(“0”);
         4. }
      8. }
   3. }
3. Test plan :
   1. The inputs choices to test this program are 366621. The student chose this because, this was the given test case. This gives adequate test coverage because the sample output is given and the out given from the function and the sample output can be compared.
4. Test Results:
   1. 



1. References:
   1. For this project the student used the project 1 A pdf file to find what the constraints of this project are and sample code to learn how to perform this project’s tasks.
   2. The grading rubric to tell how the report should be written and it’s formatting.