CSE3038 Computer Organization Project 1 Report

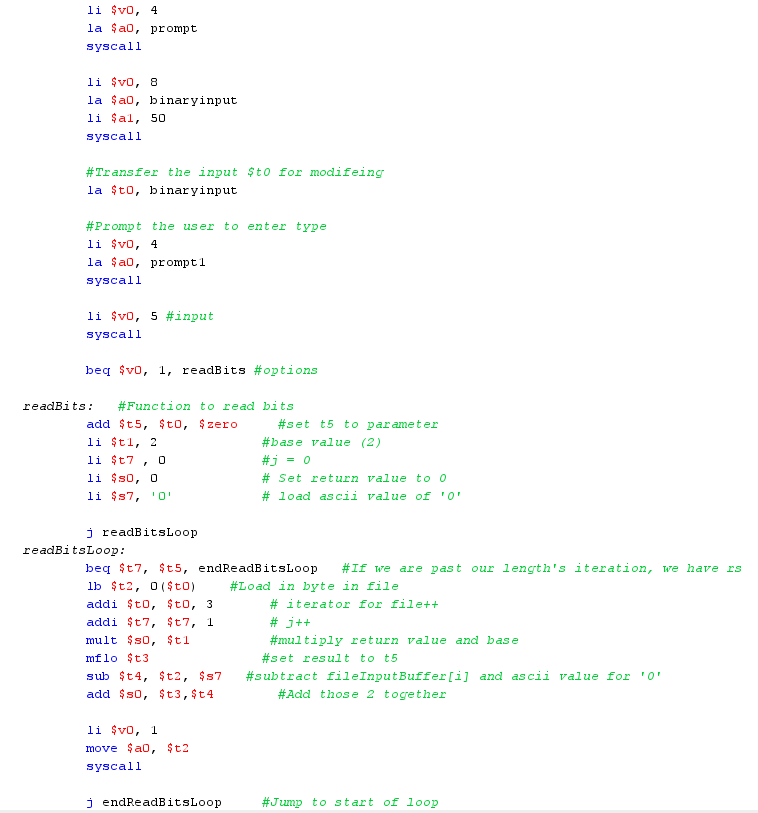
Group Members

* Ahmet Faruk Güzel 150119659
* Ahmet Aslan 150119902
* Hüseyin Kerem Mican 150119629
* Beşir Ezgin 150119780

Question 1

When we creating a binary converter, only the binary input and type that the user has entered are required as data. According to the type entered, the input 2's is converted to complement or hex.

1. With the conditional if structure, it enters the 2' complement function according to the type.
2. The 2's complement of the binary input entered in the 2's complement function is recorded in the $v0 register.



Question 2

In this question, there are 4 inputs as instructor wanted which are integer. The part shown below is the data part for this function. There are 3 delimeters to use while printing the output. metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**UI:**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

These are the values shown in project PDF. The answer is properly calculated and printed.

**Calculating steps:**

1. First we took first numerator and multiply it with second denominator for denominator equalization and did the same thing between second numerator and first denominator.
2. After that we have 10 as first numerator, 15 as second numerator. We did same simple multiplication for both denominators and get 50. After additions of numerators the number we got is 25/50, to simplify these numbers we call GCD function in our code.metin içeren bir resim

   Açıklama otomatik olarak oluşturuldu
3. In GCD function, there is a loop to find greatest common divisor to simplify numerator and denumerator. metin içeren bir resim

   Açıklama otomatik olarak oluşturuldu
4. We calculate that specific number and jump to returnnumber function which returns GCD and divides numerator and denominator and print the answer in simplified form.

Question 3

For this question, we take inputs as strings and store them. Data part is shown below.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

There are two loops to detect parsers. First loop iterates over first input which is our string to be parsed. Second loop is to search for matching character, which means if first string has a parser character we will detect it. If string has a parser character, we call newLine function to print “\n” and skip to the next line. If we reach to second strings last element which is zero, it means string is finished and also means the character is not a parser, it is a normal letter, we print it by calling print function. In newLine and print functions we increment first strings adress to reach the next element to read all of the string. When first string is finished, UI function is called to keep program working. Example in PDF shown below.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

These are the loops mentioned above:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu