ASSIGNMENT 6

ELP780 Software Lab

Kakul Shrivastava

2020EET2173

2020-2022

A report presented for the assignment on

Python Basics and Lex and Yacc

Department of Electrical Engineering

IIT Delhi

India

November 17, 2020

Table of Contents

1. Problem Statement 1

1.1 Objective :

- Given two non-zero numbers A and B, count the number of 1s in their binary representations and output if their bit balanced (number of 1s are equal) or bit biased (number of 1s are unequal)

1.2 Algorithm and Implementation :

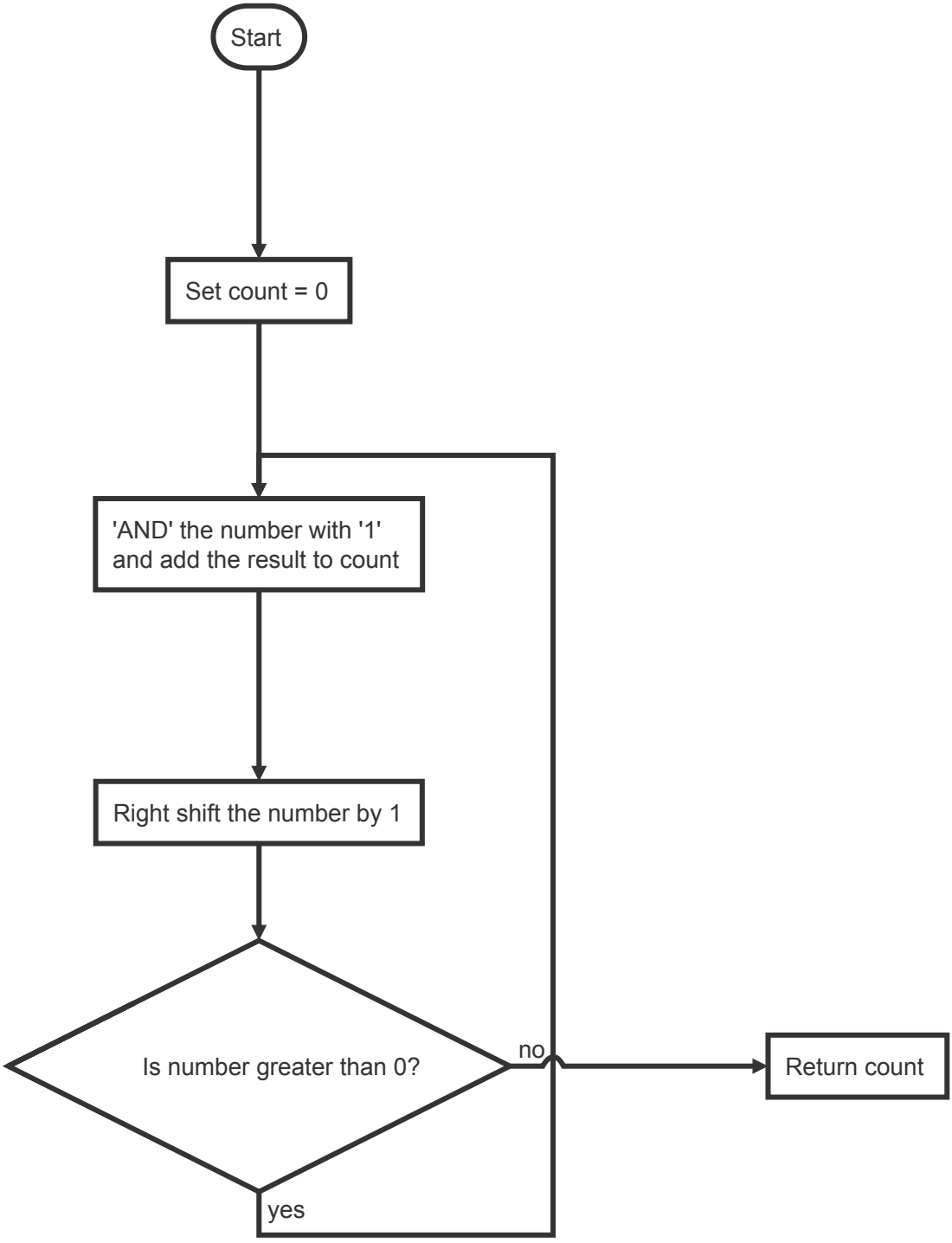
- **Function to calculate number of 1's in binary representation of a number :**
    - Initialisation : count = 0 (to store number of 1's)
    - 'AND' the number with '1' to get the LSB and add it to count (increment if it is '1' else remains same)
    - Right shift the number by 1 to get the next higher bit in the position of LSB
    - Repeat till the number is greater than 0 (has atleast one '1' in its representation)
- Take the input for both numbers
- Calculate the number of 1s in both numbers using the above function (Let the result be |A| and |B|)
- IF (|A|-|B| == 0):

  print : Bit Balanced
- ELSE :

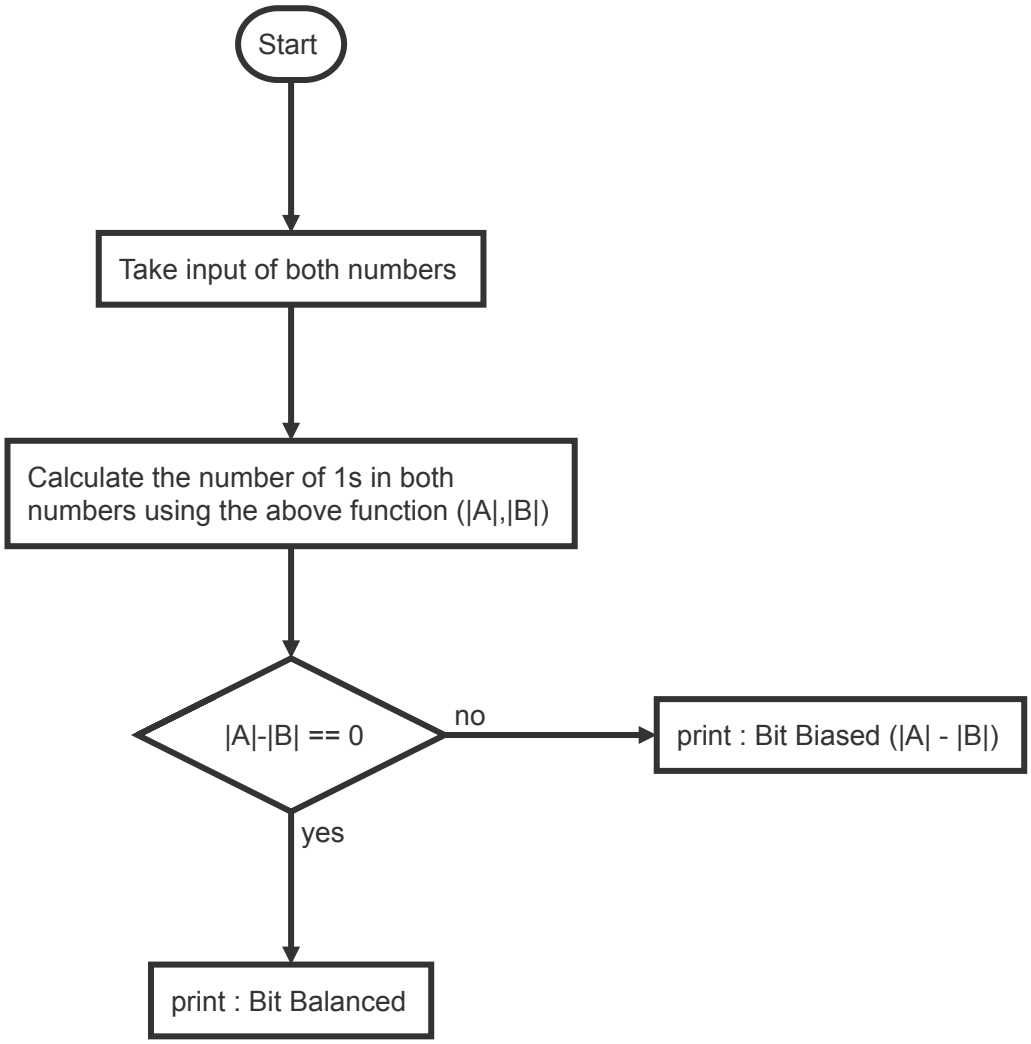
  print : Bit Biased (|A| - |B|)

1.3 Flowchart :

1.3.1 Function to calculate number of 1s in binary representation :



1.3.2 Main program :



1.4 Screenshots :

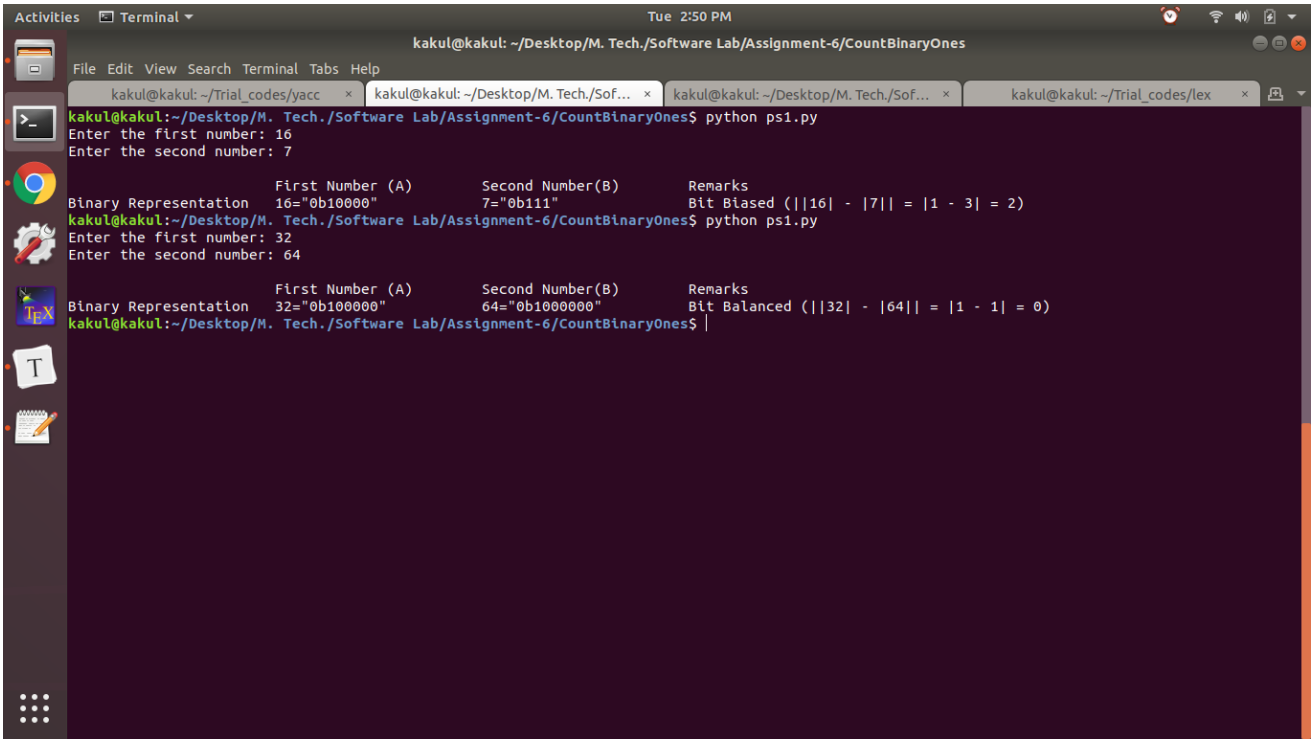


Figure 1 : Results of PS1

2. Problem Statement 2

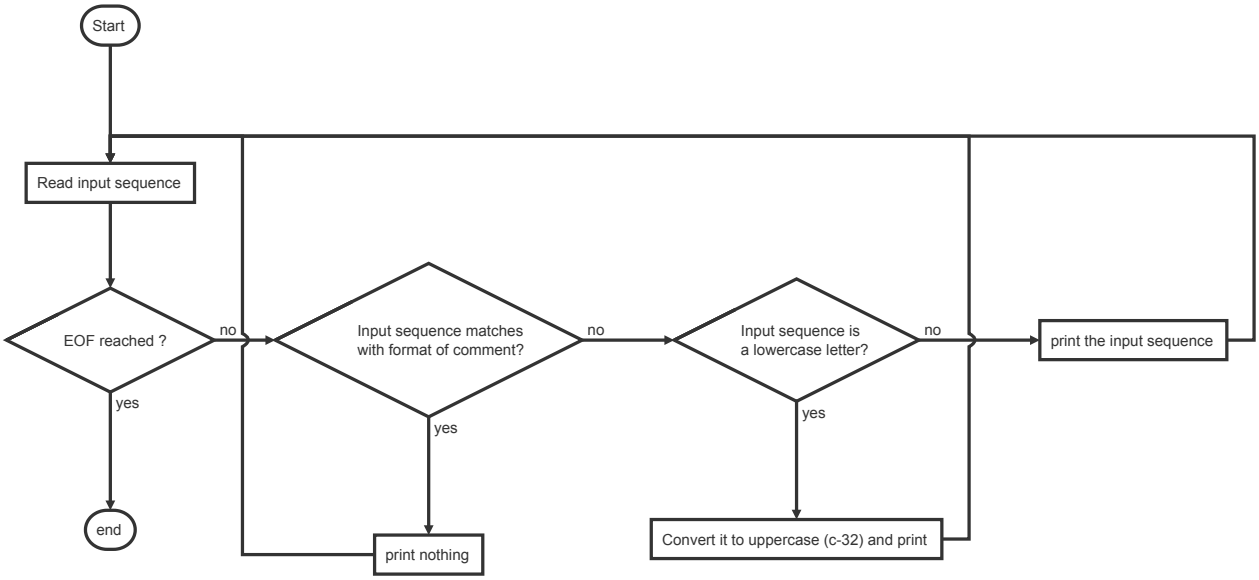
2.1 Objective :

- create a preprocessor for assembler that will translate any 8085 assembly code into full capital letters and all numerals and remove the comments

2.2 Algorithm and Implementation :

- Read the file
- IF (Input sequence matches with format of comment) :
  - print nothing
- ELSE IF (Input sequence is a lowercase letter)
  - Convert it to uppercase (c-32) and print
- ELSE
  - print the input sequence

2.3 Flowchart :



2.4 Screenshots :

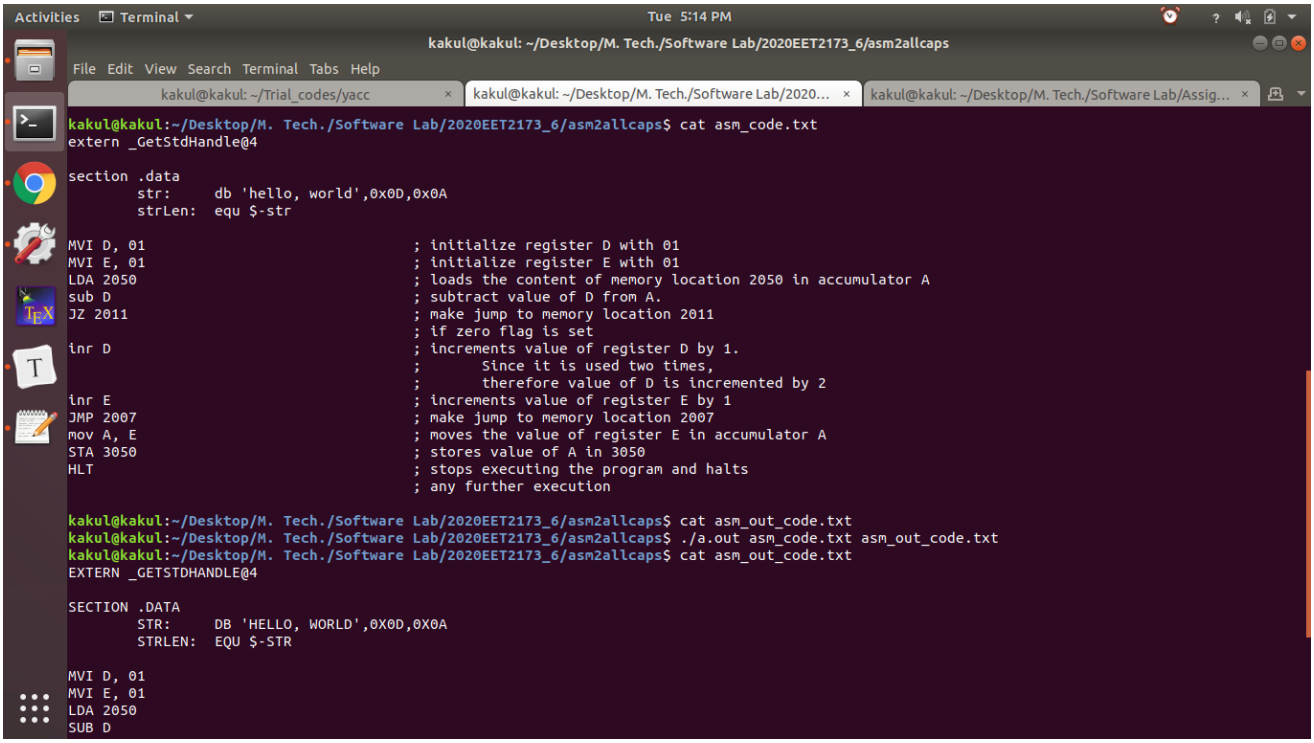


Figure 2 : Results of PS2

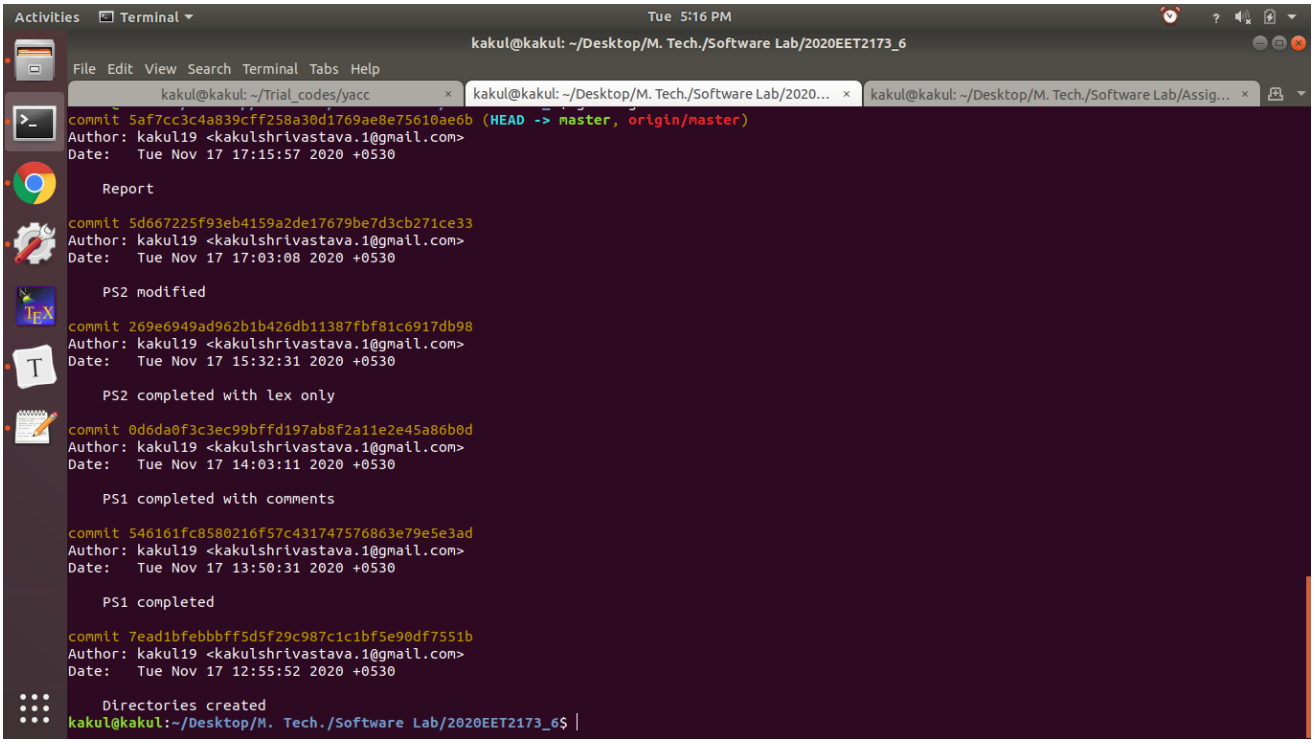


Figure 3 : Git log

3. Appendix

3.1 Code for Problem Statement 1

```
#Function to calculate number of 1's in binary representation of a number

def count_ones(n):
    count=0          #Initializing count to store number of 1's

    while(n>0):                  #Iterating till the number is greater than 0
(has atleast one '1' in its representation)
        count=count + (n & 1)    #'AND' the number with 1 to get the LSB and
add it to count (increment if it is '1' else remains same)
        n=(n>>1)                 #Right shift the number by 1 to get the next
higher bit in the position of LSB

    return count                 #Return the value of count after iteration is
complete

a=int(input("Enter the first number: "))     #Take the input of first number
b=int(input("Enter the second number: "))    #Take the input of second number

A=count_ones(a)                              #Count number of 1s in first
number
B=count_ones(b)                              #Count number of 1s in second
number

#Printing the output in desired format

print("\n\t\t\tFirst Number (A) \tSecond Number(B) \tRemarks")
print("Binary Representation \t{}=\"{}\" \t\t{}=\"{}\" \t\t{} (||{}| - |{}|| =
|{} - {}| = {})"
        .format(a,bin(a),b,bin(b),"Bit Balanced" if (A-B==0) else "Bit
Biased",a,b,A,B,abs(A-B)))
```

3.2 Code for Problem Statement 2

```
%{
 #include<stdio.h>

%}

%%
[;].* {fprintf(yyout, "");}

[a-z] {fprintf(yyout,"%c",yytext[0]-32);}

. {fprintf(yyout, "%s", yytext); }
%%

yywrap();

int main(int argc, char **argv)
{
```

```
    extern FILE *yyin, *yyout;
    yyin = fopen(argv[1], "r");
    yyout = fopen(argv[2], "w");
    yylex();
    return 0;

}
```

References

1. *Lex and Yacc: A Brisk Tutorial* , https://www2.cs.arizona.edu/~debray/Teaching/CSc453/DOCS/tutorial-large.pdf
2. *flex & bison* , http://web.iitd.ac.in/~sumeet/flex__bison.pdf