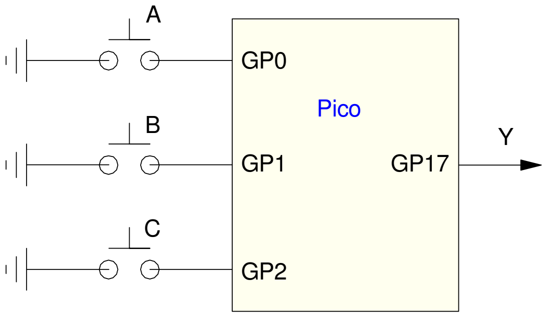
ECE 476/676 - Test #1: Name ____________________

1) Assume three push-buttons are connected to a Pi-Pico as shown. Each button's logic value is:

- 1: Button not pushed
- 0: Button pushed



Write a Python program which outputs the following logic:

- $Y = AB + \bar{B}C$

Y BC

A		00	01	11	10
	0	0	1	0	0
	1	0	1	1	1

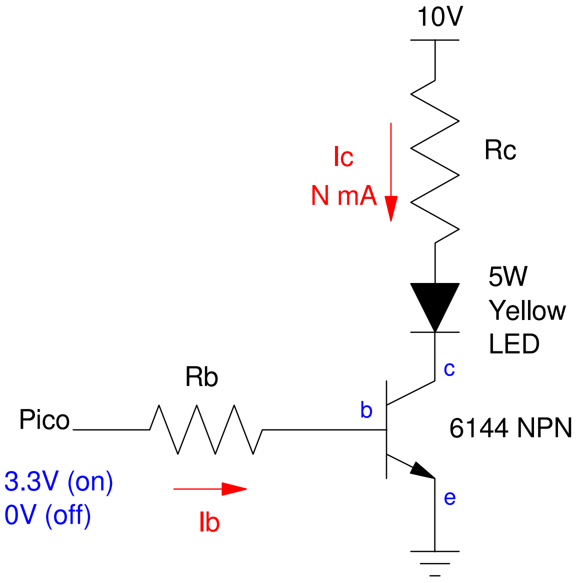
2) Digital Outputs: Determine Rb and Rc so that your Pi-Pico can drive a white 5W yellow LED at N mA where N is related to your birthday

- Vf = 2.4V @ 1200mA
- 600 Lumens @ 1200mA
- N = 900 + 100*(birth month) + (birth date).

Assume a 6144 NPN transistor

- Vbe = 700mV
- Vce(sat) = 360mV
- Current gain = = 200

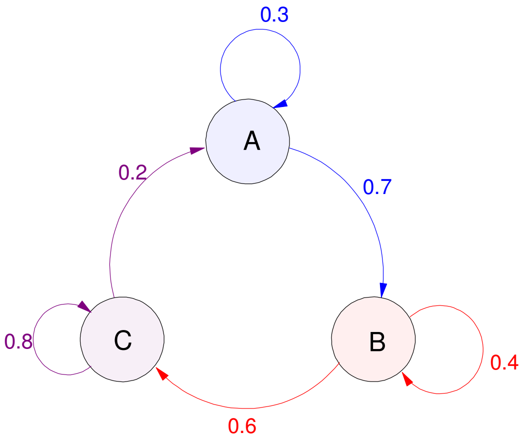
N mA 900 + 100*(Birth Month) + Birth Date ex: May 14th = 1414mA	Rb	Rc



3) Hot Potato: Write a Python program to play a game of hot potato.

- Three players are involved: A, B, C
- Player A starts with the potato
- Every 1 second, the potato is passed
- The probability that a player keeps or passes the potato right is show in the figure.
  - Example: When it's time to pass the potato, A keeps it 30% of the time and passes it on to B 70% of the time.
- The time and the person who is holding the potato is displayed in the shell window every toss (every 1 second)

```
Toss # 4     A has the potato
Toss # 5     B has the potato
```



Give a Python program to simulate this game.

Note: To generate a random number, p, in the range of (0,1), use the following code

```
from random import random
p = random()
```

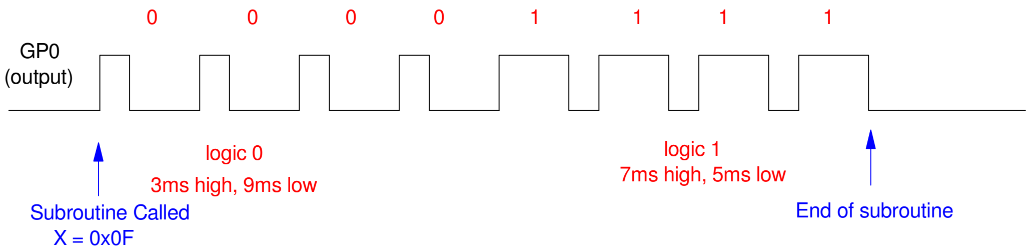
4) Bit Banging: Give a Python subroutine which is passed an 8-bit number in the range of (0..255) inclusive

The program then outputs a binary sequence based upon each bit of this number

- As 8-bit number is received (X = 0..255)
- Data is output on pin GP0
- Starting with the most significant bit of X, output a 1 or 0 based upon that bit's value
- If that bit is logic 0, output a 3ms pulse on GP0
- If that bit is logic 1, output a 7ms pulse on GP0
- To total length of each bit is 12ms

After all 8 bits are sent, line GP0 goes low and the subroutine exits

- For example, if the data passed is 0x0F (X = 0x0F = 15), the signal on GP0 should look like this:



Write the corresponding Python code

```
def Bit_Banging(X):
```

Generally Useful Python Routines

Binary Input (Button Pressed)

```
from machine import Pin

Button = Pin(15, Pin.IN, Pin.PULL_UP)
x = Button.value()
```

Binary Output (Blinking Light)

```
from machine import Pin

LED = Pin(16, Pin.OUT)
LED.toggle()
LED.value(1)
LED.value(0)
```

Analog Input (A2D Read)

```
from machine import ADC

a2d0 = ADC(0)
x = a2d0.real_u16()
```

Analog Output (PWM Output)

```
from machine import Pin, PWM

Aout = Pin(16, Pin.OUT)
Aout = PWM(Pin(16))
Aout.freq(1000)

# 0% duty cycle
Aout.duty_u16(0x0000)

# 100% duty cycle
Aout.duty_u16(0xFFFF)

# 50us pulse
Aout.duty_ns(50_000)
```

Measure a pulse width in micro-seconds

```
from machine import Pin, time_pulse_us

X = Pin(19, Pin.IN, Pin.PULL_UP)
low = time_pulse_us(19, 0, 500_000)
high = time_pulse_us(19, 1, 500_000)
```

Pause 1.23 seconds

```
from time import sleep

sleep(1.23)
```

For Loops

```
for i in range(0,6):
    d1 = i
    for j in range(0,4):
        d2 = j
        y = d1 + d2
```

While Loops

```
t = 0
while(t < 5):
    t = t + 0.01
    print(t)
```

If - else if - else statements

```
if(x < 10):
    a = 1
elif(x < 20):
    a = 2
else:
    a = 3
```

Random Numbers

```
from random import random

x = random()
# x = 0.0000 to 0.9999
```

Measure time since reset

```
from time import ticks_us

x0 = ticks_us()
```

Logic

```
&    logical and
|    logical or
^    logical xor
>>   shift right
<<   shift left
```