Name: Alan Bernal

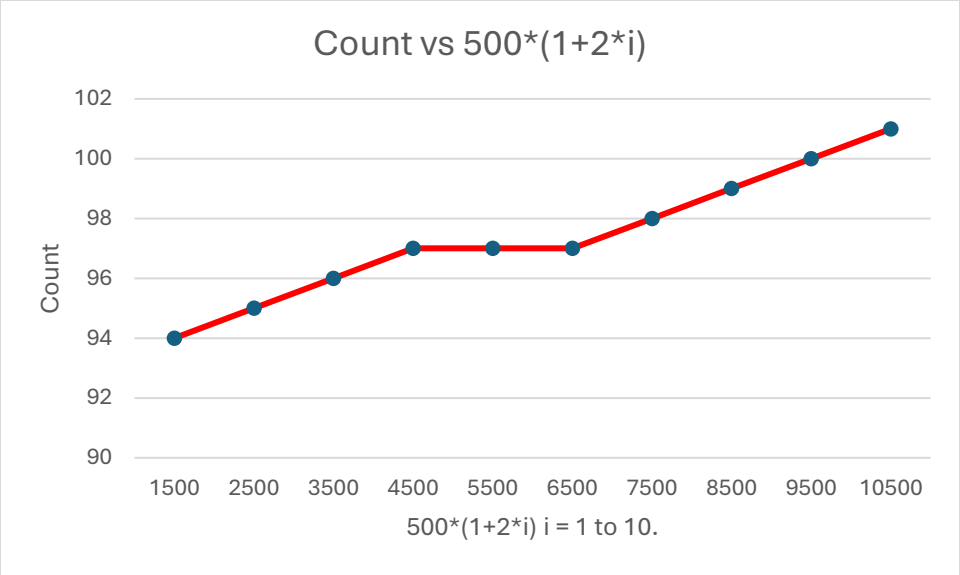
Code :

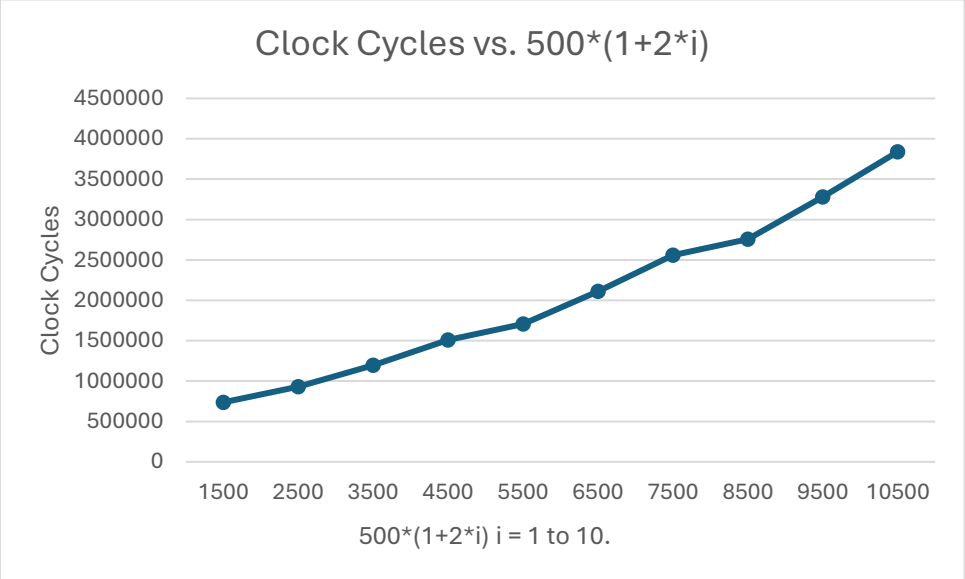
```
/*
 *  HW Assignment: CPGM0
 *  EEL-4746 Fall 2025
 *  Alan Bernal
 *  Date: 10/8/2025
 *  This is my first EEL-4746 C Program
 */

// Standard Includes

#include "driverlib.h"
#include <stdint.h>
#include <stdio.h>

//Include file for BCUART function
#include "HAL_UART_4746.h"

// Function Prototypes
void GPIO_init();
bool isPrime(uint16_t n);


//  Main Function
void main(void){

   //Define Local Variable
   uint16_t count;
   uint16_t i, x;
   char buffer[100];


   //WDT
   WDT_A_hold(WDT_A_BASE);
```

```
//Initialize LED0 and set it low
GPIO_setAsOutputPin(GPIO_PORT_P1, GPIO_PIN0);
GPIO_setOutputLowOnPin(GPIO_PORT_P1, GPIO_PIN0);

//Initialize and Configure UART
UART_initGPIO();
UART_init();

//Activate New Port Configurations
PMM_unlockLPM5();

//Writing your name
sprintf(buffer, "Your Name: Alan Bernal \r\n");
UART_transmitString(buffer);

//Writing todays date
sprintf(buffer, "Today's date is 10, 8, 2025\r\n");
UART_transmitString(buffer);

//Writing Course Section
sprintf(buffer, "My Course Section is Section 0001 \r\n");
UART_transmitString(buffer);

//Turn LED ON.. done
GPIO_setOutputHighOnPin(GPIO_PORT_P1, GPIO_PIN0);



//CODE

for(i = 0; i < 11; i++){
  for(x = 1; x <= 500; x++){
    if(isPrime(x*(1+2*i))){
      count++;
    };
  };
  sprintf(buffer, "%d\r\n", count);
    UART_transmitString(buffer);

};

sprintf(buffer, "%d\r\n", count);
```

```
  UART_transmitString(buffer);


  //Spin Loop
  while(1){
    // Nothing here.
  }


}

bool isPrime(uint16_t n){
  uint16_t j;
  if(n <= 1) return false;
  for(j = 2; j * j <= n; j++) {
    if(n % j == 0) return false;
  }
  return true;
};
```

i	500*(1+2*i)	Count	Clock Cycles
1	1500	94	736749
2	2500	95	932248
3	3500	96	1194073
4	4500	97	1508956
5	5500	97	1705377
6	6500	97	2111356
7	7500	98	2558461
8	8500	99	2755171
9	9500	100	3278326
10	10500	101	3839019


Count vs 500*(1+2*i)
Count
102
100
98
96
94
92
90
1500
2500
3500
4500
5500
6500
7500
8500
9500
10500
500*(1+2*i) i = 1 to 10.


Clock Cycles vs. 500*(1+2*i)
Clock Cycles
4500000
4000000
3500000
3000000
2500000
2000000
1500000
1000000
500000
0
1500
2500
3500
4500
5500
6500
7500
8500
9500
10500
500*(1+2*i) i = 1 to 10.