LAB 1

Aniket Sambher

Reg no-190905466

Roll no -58

1.Euclid

Algo:

Step 1 If n = 0, return m and stop; otherwise go to Step 2

Step 2 Divide m by n and assign the value of the remainder to r

Step 3 Assign the value of n to m and the value of r to n. Go to

Step 1.

while n ≠ 0 do

r ← m mod n

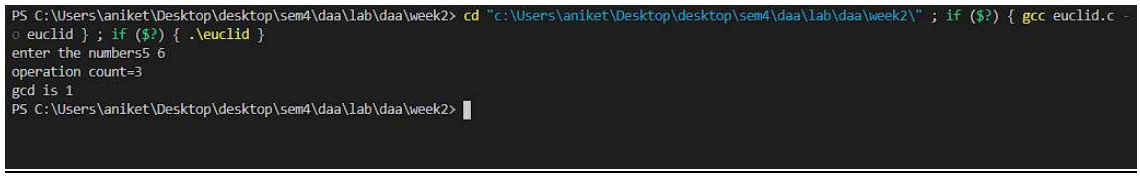
m← n

n ← r

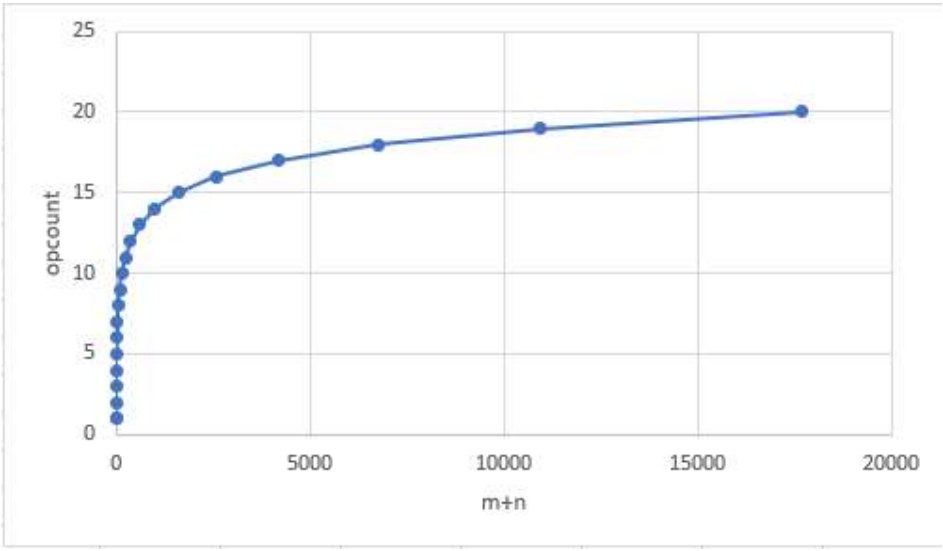
return m

```
#include<stdio.h>
#include <math.h>
unsigned int euclid(unsigned int m,unsigned int n) {
   unsigned int r;
   int opcount=0;
   while(n!=0){
      opcount++;
      r=m%n;
      m=n;
      n=r;

   }
   printf("operation count=%d\n",opcount);
   return m;
```

```
}

void main(){
   int a,b;
   printf("enter the numbers");
   scanf("%d%d",&a,&b);
         int gcd=euclid(a,b);
         printf("gcd is %d",gcd);









         }
```



m+n	euclid opcount	
1	1	
2	1	
3	2	
5	3	
8	4	
13	5	
21	6	
34	7	
55	8	
89	9	
144	10	
233	11	
377	12	
610	13	
987	14	
1597	15	
2584	16	
4181	17	
6765	18	
10946	19	
17711	20	



2.**consecutive integer**

Consecutive integer checking algorithm

Step 1 Assign the value of min{m,n} to t

Step 2 Divide m by t. If the remainder is 0, go to Step 3;

otherwise, go to Step 4

Step 3 Divide n by t. If the remainder is 0, return t and stop;

otherwise, go to Step 4

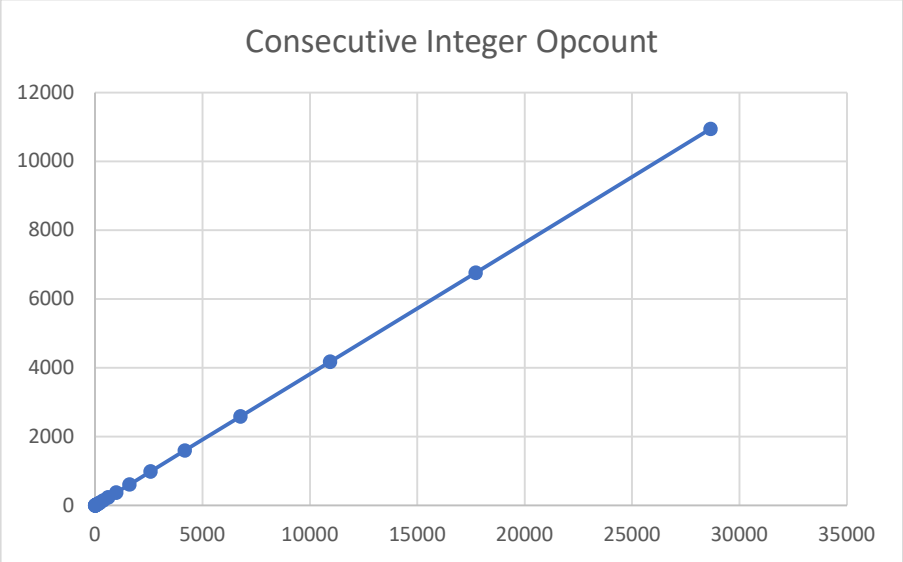
Step 4 Decrease t by 1 and go to Step 2

```
#include <stdio.h>
#include <stdlib.h>

int opcount = 0; // variable to count how many times the basic operation executes.



int gcd(int m, int n) {
   int t = m > n ? n: m;

   while(m % t != 0 || n % t != 0) {
     opcount++;
     t--;
   }

   return t;
}

int main() {
            int i;
            int a, b;

   printf("Enter the two numbers whose GCD has to be calculated : \n");
            scanf("%d%d", &a, &b);
```

```
        int g = gcd(a, b);
    printf("\nOperation count= %d\n", opcount);
        printf("GCD = %d\n", g);
}
```

```
PS C:\Users\aniket\Desktop\desktop\sem4\daa\lab\daa\week2> cd "c:\Users\aniket\Desktop\desktop\sem4\daa\lab\daa\week2\" ; if ($?) { gcc consecutiv
e.c -o consecutive } ; if ($?) { .\consecutive }
Enter the two numbers whose GCD has to be calculated :
2 3

Operation count= 1
GCD = 1
PS C:\Users\aniket\Desktop\desktop\sem4\daa\lab\daa\week2>
```

A	B
m+n	Consecutive Integer Opcount
1	0
2	0
3	0
5	1
8	2
13	4
21	7
34	12
55	20
89	33
144	54
233	88
377	143
610	232
987	376
1597	609
2584	986
4181	1596
6765	2583
10946	4180
17711	6764
28657	10945



3. middle school

Middle-school procedure

Step 1 Find the prime factorization of m

Step 2 Find the prime factorization of n

Step 3 Find all the common prime factors

Step 4 Compute the product of all the common prime factors

and return it as gcd(m,n)

```
#include<stdio.h>
#include<math.h>
int opcount = 0;
int primeFactors(int arr[], int n){

        int k = 0;
```

```
        while(n%2 == 0){
                arr[k++] = 2;
                n = n/2;
                opcount++;
        }
        for(int i = 3; i<n; i=i+2){
                opcount++;
                while (n%i==0)
                        {arr[k++] = i;
                        n = n/i;
                        opcount++;
                }
        }
        if(n>2){
        arr[k++] = n;
        opcount++;
        }
        return k;
}
int gcdMidSchool(int m, int n){
        if(m==0 && n==0){
        return -1;
        }
        if(m==0|| n==0 ){
        opcount = 1;
        return m+n;
        }
        int mArr[m],nArr[n];
        int k1,k2,prod;
 // k3 = 0;
        prod = 1;
```

```
k1 = primeFactors(mArr,m);
k2 = primeFactors(nArr,n);
if(k1<k2){
	for(int i = 0; i<k1; ++i){
		for(int j = i; j<k2;){
			opcount++;
			if(mArr[i] == nArr[j]){
				prod*=mArr[i];
				++i;
				++j;
				continue;
			}
			++j;
		}
	}
}
else{
	for(int i = 0; i<k2; ++i){
		for(int j = i; j<k1;){
			opcount++;
			if(nArr[i] == mArr[j]){
				prod*=nArr[i];
				++i;
				++j;
				continue;
			}
			++j;
		}
	}
}
```

```
        return prod;
}
int main(){
        int m,n;
        printf("Enter m and n - ");
        scanf("%d %d",&m,&n);
        int gcd = gcdMidSchool(m,n);
        if(gcd == -1){
                printf("Undefined gcd");
        }
        else{
                printf("GCD is %d",gcd);
                printf("\nOpcount is : %d", opcount);
        }
 return 0;
}
```

```
PS C:\Users\aniket\Desktop\desktop\sem4\daa\lab\daa\week2> cd "c:\Users\aniket\Desktop\desktop\sem4\daa\lab\daa\week2\" ; if ($?) { gcc middlescho
ol3.c -o middleschool3 } ; if ($?) { .\middleschool3 }
Enter m and n - 5 6
GCD is 1
Opcount is : 6
```