



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

DESIGN AND ANALYSIS OF ALGORITHMS

NAME OF STUDENT :- ANIKET SHARMA

REGISTRATION NO. :- RA1911003030436

BRANCH/DEGREE :- B.TECH/CSE

SUBJECT CODE :-18CSC204J

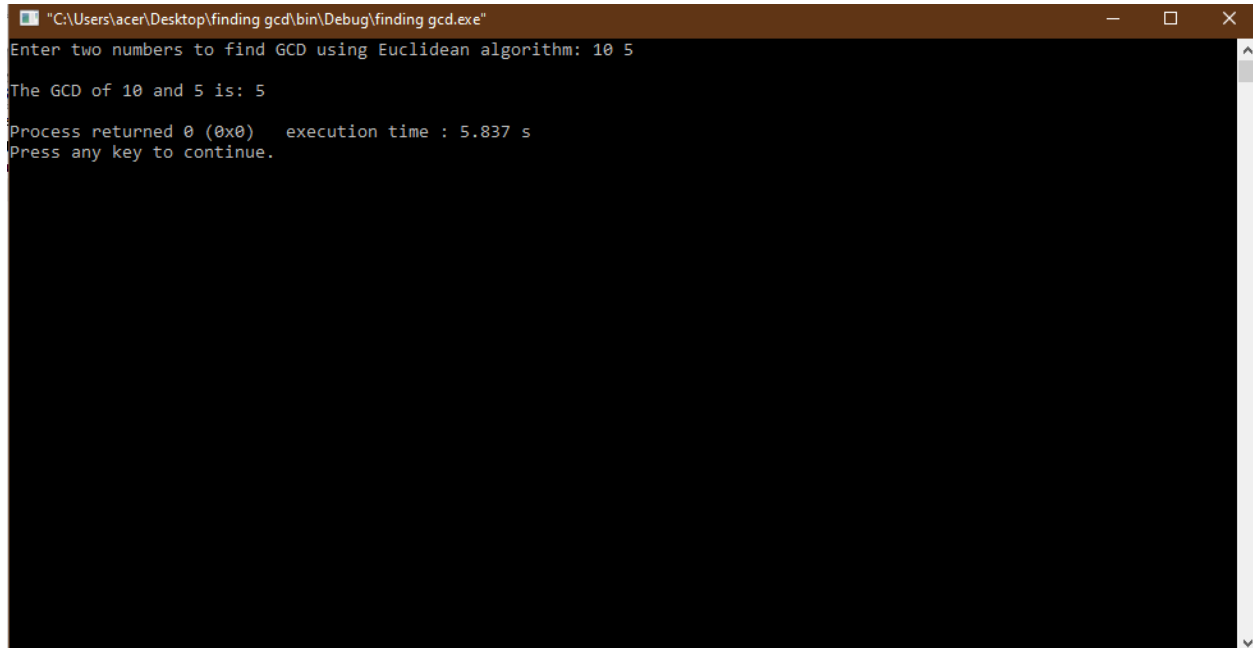
SEMESTER/SECTION :- 4TH SEM, I

FACULTY INCHARGE :- Himansu Sekhar
Pattanayak

USING 1st method:-

```
1  #include<iostream>
2  #include<conio.h>
3  #include<stdlib.h>
4
5  using namespace std;
6  int gcd(int u, int v)
7  {
8      return (v != 0) ? gcd(v, u % v) : u;
9  }
10
11  int main(void)
12  {
13      int num1, num2, result;
14      cout << "Enter two numbers to find GCD using Euclidean algorithm: ";
15      cin >> num1 >> num2;
16      result = gcd(num1, num2);
17      if (gcd)
18          cout << "\nThe GCD of " << num1 << " and " << num2 << " is: " << result
19              << endl;
20      else
21          cout << "\nInvalid input!!!\n";
22      return 0;
23  }
24
```

OUTPUT:-



```
"C:\Users\acer\Desktop\finding gcd\bin\Debug\finding gcd.exe"
Enter two numbers to find GCD using Euclidean algorithm: 10 5

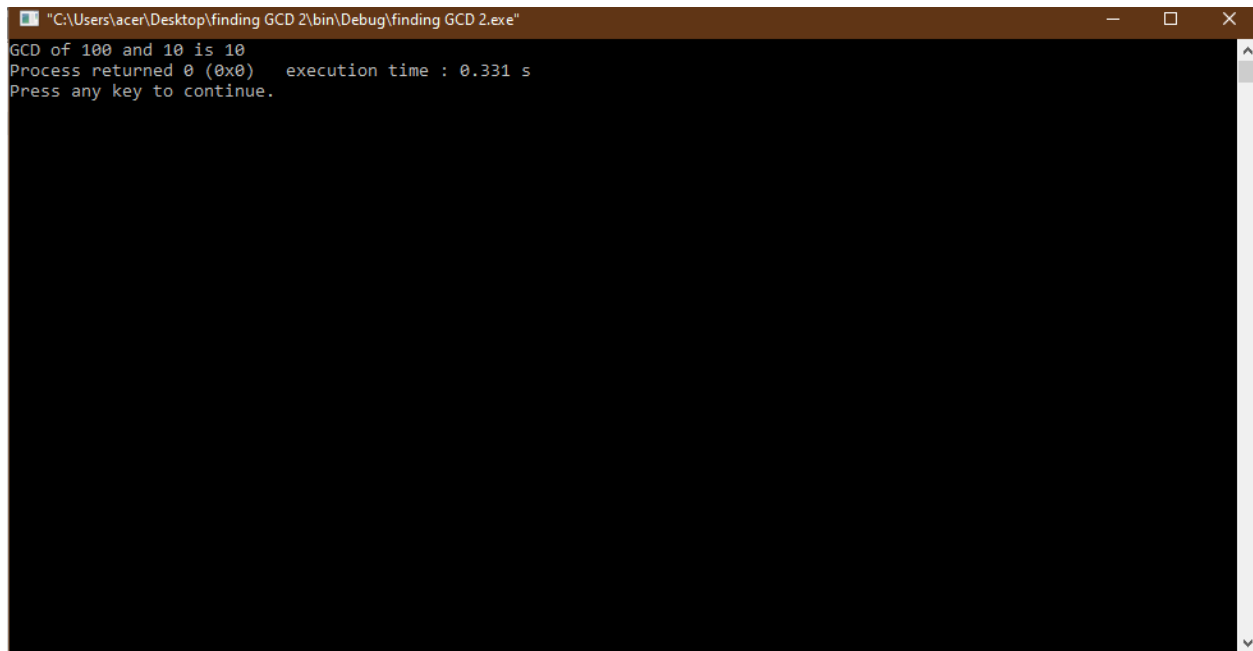
The GCD of 10 and 5 is: 5

Process returned 0 (0x0)   execution time : 5.837 s
Press any key to continue.
```

USING 2ND METHOD:-

```
1  #include <iostream>
2  using namespace std;
3  int gcd(int a, int b)
4  {
5
6      if (a == 0)
7          return b;
8      if (b == 0)
9          return a;
10
11
12      if (a == b)
13          return a;
14
15
16      if (a > b)
17          return gcd(a-b, b);
18      return gcd(a, b-a);
19  }
20
21
22  int main()
23  {
24      int a = 100, b = 10;
25      cout<<"GCD of "<<a<<" and "<<b<<" is "<<gcd(a, b);
26      return 0;
27  }
28
```

OUTPUT:-



```
"C:\Users\acer\Desktop\finding GCD 2\bin\Debug\finding GCD 2.exe"
GCD of 100 and 10 is 10
Process returned 0 (0x0)   execution time : 0.331 s
Press any key to continue.
```

USING 3RD METHOD:-

```
1  #include <bits/stdc++.h>
2  #define MAXFACTORS 1024
3  using namespace std;
4  //structure to store factorization
5  typedef struct{
6      int size;
7      int factor[MAXFACTORS + 1];
8      int exponent[MAXFACTORS + 1];
9  } FACTORIZATION;
10 void FindFactorization(int x, FACTORIZATION* factorization){
11     int i, j = 1;
12     int n = x, c = 0;
13     int k = 1;
14     factorization->factor[0] = 1;
15     factorization->exponent[0] = 1;
16     for (i = 2; i <= n; i++) {
17         c = 0;
18         while (n % i == 0) {
19             c++;
20             n = n / i;
21         }
22         if (c > 0) {
23             factorization->exponent[k] = c;
24             factorization->factor[k] = i;
25             k++;
26         }
27     }
28     factorization->size = k - 1;
29 }
30 //printing the factors
31 void DisplayFactorization(int x, FACTORIZATION factorization){
32     int i;
33     cout << "Prime factor of << x << = ";
34     for (i = 0; i <= factorization.size; i++) {
35         cout << factorization.factor[i];
36         if (factorization.exponent[i] > 1)
37             cout << "^" << factorization.exponent[i];
38         if (i < factorization.size)
39             cout << " * ";
40         else
41             cout << "\n";
42     }
43 }
44
45 int gcd(int m, int n){
46     FACTORIZATION mFactorization, nFactorization;
47     int r, m1, n1, i, k, x = 1, j;
48     FindFactorization(m, mFactorization);
49     DisplayFactorization(m, mFactorization);
50     FindFactorization(n, nFactorization);
51     DisplayFactorization(n, nFactorization);
52     int min;
53     i = 1;
54     j = 1;
55     while (i <= mFactorization.size && j <= nFactorization.size) {
56         if (mFactorization.factor[i] < nFactorization.factor[j])
57             i++;
58         else if (nFactorization.factor[j] < mFactorization.factor[i])
59             j++;
60         else{
61             min = mFactorization.exponent[i] > nFactorization.exponent[j] ? nFactorization.exponent[j] : mFactorization.exponent[i];
62             x = x * mFactorization.factor[i] * min;
63         }
64     }
65     return x;
66 }
```

```

62         x = x * mFactorization.factor[i] * min;
63         i++;
64         j++;
65     }
66 }
67 return x;
68 }
69 int main(){
70     int m = 48, n = 36;
71     cout << "GCD("<< m << ", " << n << ") = " << gcd(m, n);
72     return (0);
73 }
74

```

OUTPUT:-

```

"C:\Users\acer\Desktop\Finding gcd 3\main.exe"
Prime factor of << x << = 1*2^4*3
Prime factor of << x << = 1*2^2*3^2
GCD(48, 36) = 12
Process returned 0 (0x0)   execution time : 0.323 s
Press any key to continue.

```