/9/25

Task - 6

Basic Number Theory - 2

Aim :- To write and execute the program based on basic Number Theory 2.

Algorithm

1. Read the values of x_1, x_2, y_1 and y_2.

2. Calculate the difference between x_2 and x_1 and store it in a Variable dx.

3. Calculate the difference between y_2 and ~~x~~ y_1 and store it in Variable dy.

4. Calculate the square of dx and store it in a variable dx_2

5. Calculate the square of dy and store it in a variable dy_2

6. Calculate the sum of dx_2 and dy_2 and store it in a Variable d_2.

7. Calculate the Square root of d_2 and store it in a Variable distance.

8. Print the value of distance as output.

9. End the program.

and Task 6-Basic Number
Theory - 2

1. **Program:**

```
nclude <stdio.h>
nclude <math.h>

main() { int x1,
y1,    x2,    y2;
double distance;

printf("Enter x1 and y1: ");
scanf("%d    %d",    &x1,
&y1); printf("Enter x2 and
2: ");
canf("%d %d", &x2, &y2);

istance = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));

intf("The Euclidean distance between (%d, %d) and (%d, %d) is %lf\n", x1, y1, x2, y2, distance);
turn 0;
```

Task - 6B

GCD

Algorithm

1. Read the Value of N and the array of integers array[] of size N.
2. Read the value
3. Initialize to a Variable max_len to 0, which will store the length of the largest contiguous subarray whose GCD is at least k.
4. Initialize two variables, left and right, to 0, which represent the GCD of the current subarray.
5. Initialize a Variable current_gcd to array[0],
6. Start a loop with Variable i from N-1 :
   a. update the current_gcd to be GCD of current-gcd and array[i]
   b. while the current_gcd is greater than or equal.
7. If max_len is still 0, print "0" as there is no subarray whose GCD is atleast k.

8. End the program.

Program:

```
 #include  <stdio.h>
 int gcd(int a, int b) {
 if (b == 0) { return
 a; } else {

  return gcd(b, a % b);
    }
 }

int main() { int
   n, k;
   scanf("%d %d", &n, &k);

   int arr[n];
   for (int i = 0; i < n; i++) { scanf("%d",
      &arr[i]);
   }

  int maxLength = 0; int
  length = 0;
  for (int i = 0; i < n; i++) {
     int currentGcd = arr[i];
     if (currentGcd >= k) {
     length = 1;
     } else { continue;
     }

    for (int j = i + 1; j < n; j++) { currentGcd
       = gcd(currentGcd, arr[j]);
       if (currentGcd >= k) {
          length++;
       } else { break;
       } }

   if (length > maxLength) {
      maxLength = length;
   }
}

printf("%d\n", maxLength);
return 0; }
```

V I I. TECH - CSE	
EX NO	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	3
VIVA VOCE (3)	3
RECORD (4)	4
TOTAL (15)	15
SIGN WITH DATE	

sult :- Thus, the program is executed and verified successfully.