**Lab Practicals – Phase – 1**

**Execute the following programs using gmp library in C , as well as in Python.**

**Note:** Do not use predefined functions from any Library or Header file, as far as possible. Instead write your own user define function for it.

**Abbreviation:** HAC- Handbook of Applied Cryptography

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Program** | **Reference, if any** |
| 1. | [Common divisors] Given integers, print all common divisors of , in ascending order.  e.g i/p: 3 12 18 36  o/p: 1 2 3 6 (i.e. a common divisors of 12, 18, 36, the number inputs n=3) |  |
| 2. | Extended Euclidean algorithm to output when is given, such , where x<y. | Notes or Algo 2.107 HAC |
| 3. | [Fundamental Theorem of Arithmetic] Given any integer output the product of primes, in ascending order. If same prime appear multiple times then print it that many times with spaces in between. | Notes |
| 4. | [Reduced Residue System Modulo m] Given an integer m, output the RRSM\_m set of integers. And also output the value of , at end. | Notes |
| 5. | Given and , output . Use fermat theorem concept. Also print intermediate equations while computing, in any readable form. | Algo 2.143 HAC |
| 6. | Given a and m, first print whether multiplicative inverse of exist Y/N then output its inverse, if exist. | Algo 2.142 HAC |
| 7. | [Solutions of congruence] Given a, b and m, print whether solution to the congruence exist Y/N. If Yes then output the number of solutions and all the solutions. |  |
| 8. | [Solution to system of congruences] Given set of integers , print whether there exists common solution which satisfy the system of congruences of the form . If exist, then print all the solutions. Use user defined CRT function. | Notes or Algo 14.71 HAC |
| 9. | [Order] Given a and m, print order of a under modulo m. | Notes |
| 10. | [Primitive roots] Given m, print the total number of primitive roots exist and print all the primitive roots, under modulo m. | Notes |

**Instructions:**

**Name convention:** All programs should be name as prg*i\_rollno*.c or prig*\_rollno*.py, if it is program 1 then prg1\_bt19cse001.c or prg1\_bt19cse001.py

All input to the program should be command line input. We test it using command line input only. For example, for prg2\_rollno.c say its executable file name is prg2\_rollno then we test as,

$.\prg2\_rollno 5 7 <enter>

Where $ is command prompt, a = 5 and b = 7, as per prg2\_rollno.c question.

And print only exact output (without any English sentences). Because we store output in a text file using linux command, such as $.\prg2\_rollno 5 7 >> out.txt <enter>

We also do **plagiarism** test of each program, if percentage of similarity is higher (around 70%) then marks will not be given. Actual percentage fixation will be decided later.