International Journal of Research in Engineering, Science and Management
Volume-1, Issue-12, December-2018
www.ijresm.com | ISSN (Online): 2581-5792

Algorithms for Number Theoretic Functions and Special Numbers

Shubham Agarwal[1], Anand Singh Uniyal[2]
[1]*Associate Professor, Department of Mathematics, New Delhi Institute of Management, Delhi, India*
[2]*Professor, Department of Mathematics, Govt. Degree College, Champawat, India*

***Abstract*: Algorithms are the heart of computer science, and the subject has countless practical applications as well as intellectual depth. Today number-theoretic algorithms are used widely, due in part to the invention of cryptographic schemes based on large prime numbers. The feasibility of these schemes rests on our ability to find large primes easily, while their security rests on our inability to factor the product of large primes. In this paper we have generated new algorithms for number theoretic functions & special numbers which are well defined in a given range and also designed the flow chart for these functions & numbers.**

***Keywords*: Algorithm, Number theoretic functions, Special numbers.**

1. Introduction

Apostol, Tom M. [1, 2] defined that an arithmetic, arithmetical, or number-theoretic function is a real or complex valued function f(n) defined on the set of natural numbers (i.e., positive integers) that "expresses some arithmetical property of n" [3]. There exist many number theoretic functions, which includes Divisor function τ(n) [4], Sigma function σ(n) [4], Euler phi function ϕ(n) [5] and Mobius function μ(n) [5]. We have also studied about some special numbers like perfect numbers, twin primes and pythagorean numbers. Agarwal, S. & Uniyal, A.S. [6] designed an efficient encryption/decryption algorithm using prime weighted graph in cryptographic system for secure communication. Agarwal, S. & Uniyal, A.S. [7] generated the algorithm to find the value of Mobius function for any positive number n and define some properties of Mobius (α,β) function. Agarwal, S. & Uniyal, A.S. [8] defined multi-dimensional tree and proposed an encryption algorithm using multi-dimensional tree in public key cryptography for security of ATM password. Agarwal, S. & Uniyal, A.S. [9] found some results on number theoretic functions. Radha K., Balaji G. & Jaya Sudha VP [10] perceived that every Number Theory tool plays an important role in providing security for transmitting messages and calculated the encryption and decryption of messages by Mod. H. W. Lenstra, JR. [11] discussed the basic problems of algorithmic algebraic number theory and emphasized on the aspects that are of interest from a purely mathematical point of view, and practical issues are largely disregarded. He also showed that the study of algorithms not only increases our understanding of algebraic number fields but also stimulates our curiosity about them. Here we have generated the algorithm and flow chart for number theoretic functions and special numbers.

2. Algorithm for divisor function τ(n)

- Step 1: Start
- Step 2: C = 0
- Step 3: Input n
- Step 4: Repeat for i = 1 to n
- Step 5: If n ≡ 0 (mod i) then C ← C+1
- Step 6: End if
- Step 7: End of loop of Step 3
- Step 8: Print C
- Step 9: Stop

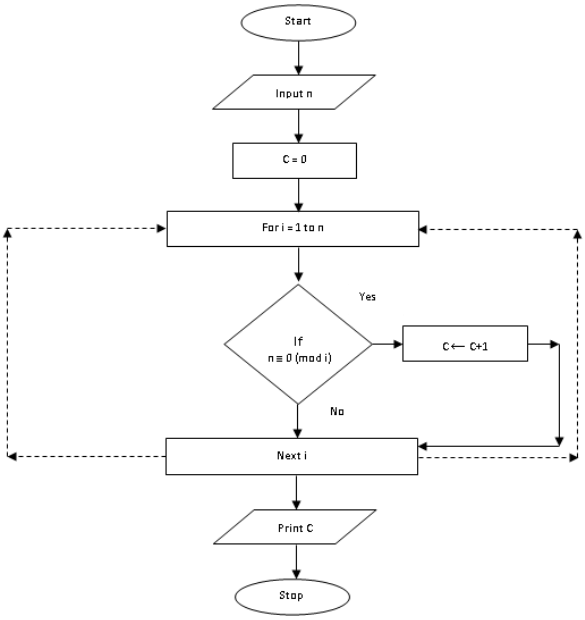

Fig. 1. . Flow chart for divisor function

3. Algorithm for sigma function σ(n)

- Step 1: Start
- Step 2: C = 0
- Step 3: Input n
- Step 4: Repeat for i = 1 to n
- Step 5: If n ≡ 0 (mod i) then C ← C+i

- Step 6: End if
- Step 7: End of loop of Step 3
- Step 8: Print C
- Step 9: Stop

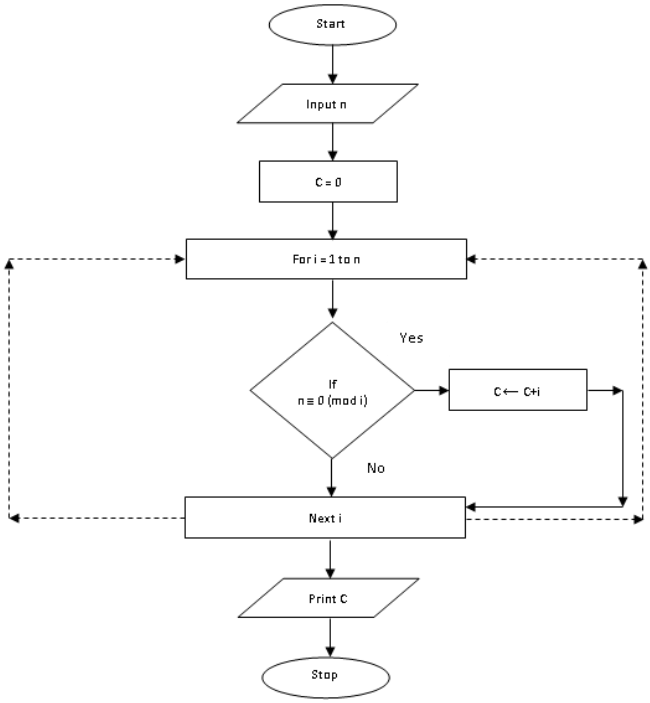

Fig. 2. Flow chart for sigma function

4. Algorithm for $\sigma_k(n)$ function

- Step 1: Start
- Step 2: Input n,k
- Step 3: Sum = 0
- Step 4: Repeat for i = 1 to n
- Step 5: If n ≡ 0 (mod i) then
- Step 6: Sum ← Sum+(int)Math.pow(i,k)
- Step 7: End if of Step 5
- Step 8: End of loop of Step 4
- Step 9: Print Sum
- Step 10: Stop

5. Algorithm for the function $\phi\ ^{\beta}\alpha(n)$

- Step 1: Start
- Step 2: C = 0
- Step 3: Input n, a, b
- Step 4: If b > n then b = n
- Step 5: End if
- Step 6: Repeat for i = a to b
- Step 7: Flag = True
- Step 8: Repeat for j = 1 to i
- Step 9: If i ≡ 0 (mod j) and n ≡ 0 (mod j)
- Step 10: Flag = False
- Step 11: Go to Step 14
- Step 12: End if
- Step 13: End of loop of Step 8
- Step 14: If Flag = True then
- Step 15: C ← C+1
- Step 16: End if
- Step 17: End of loop of Step 6
- Step 18: Print C
- Step 19: Stop

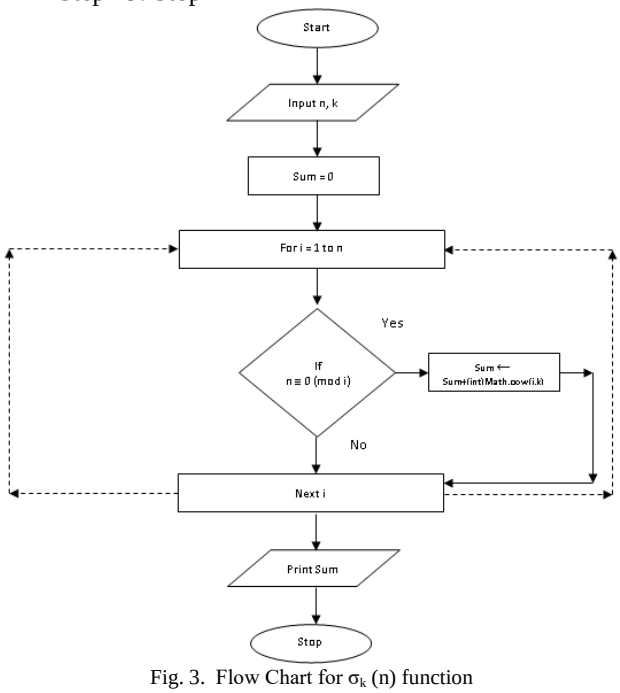

Fig. 3. Flow Chart for σ_k (n) function

6. Algorithm for mobius (α,β) function

- Step 1: Start
- Step 2: C = 0
- Step 3: Flag = True
- Step 4: Declare A[100]
- Step 5: Input n, a, b
- Step 6: p = n
- Step 7: If p = 1 then
- Step 8: Print "Mobius(1) = 1"
- Step 9: Go to Step 38
- Step 10: Else
- Step 11: Repeat Step 12 to Step 21 while n > 1
- Step 12: Div = False
- Step 13: Repeat for i = a to b
- Step 14: If n ≡ 0 (mod i) then Div = True
- Step 15: Go to Step 18
- Step 16: End if of Step 14
- Step 17: End of loop of Step 13
- Step 18: If Div = True then A[C] = i
- Step 19: End if of Step 18
- Step 20: C ← C+1
- Step 21: n = n/i

- Step 22: End of loop of Step 11
- Step 23: Repeat for i = 1 to C-2
- Step 24: Repeat for j = i+1 to C-1
- Step 25: If A[i] = A[j] then
- Step 26: Flag = False
- Step 27: Go to Step 30
- Step 28: End if of Step 25
- Step 29: End of loop of Step 24
- Step 30: End of loop of Step 23
- Step 31: If Flag = True then
- Step 32: r = (int) power (-1,c)
- Step 33: Print r
- Step 34: Go to Step 38
- Step 35: Else
- Step 36: Print 0
- Step 37: End if of Step 31
- Step 38: Stop

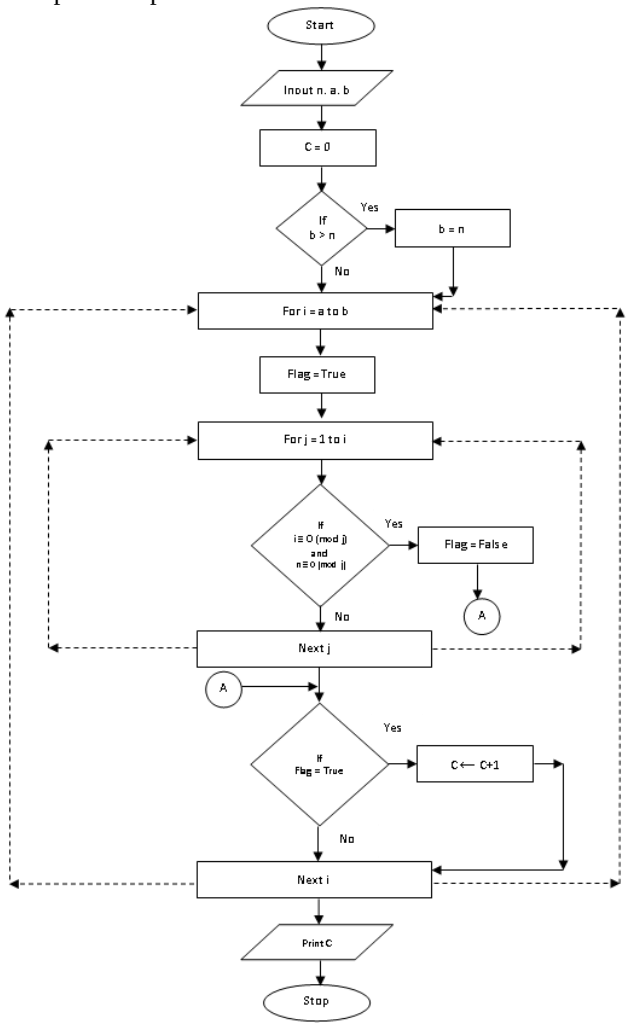

Fig. 4. Flowchart for the function $\phi^{\beta}\alpha(n)$

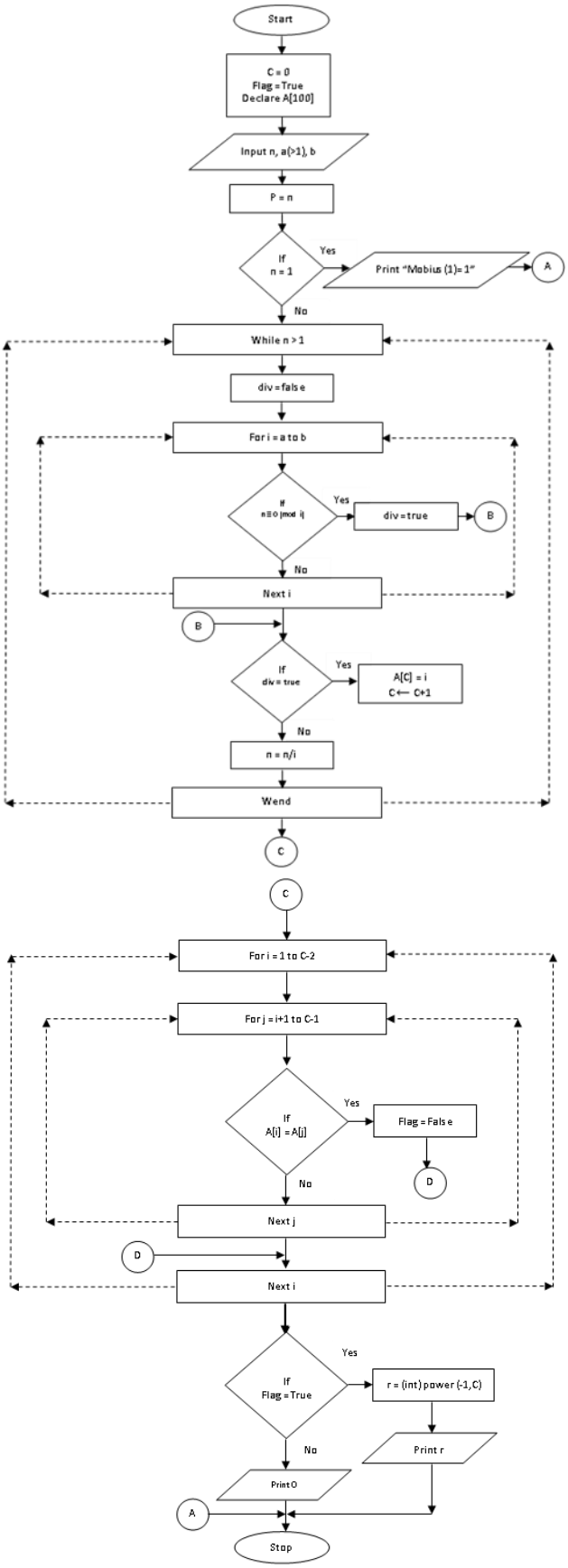

Fig. 5. Flow Chart for Mobius (α,β) function

7. Algorithm for perfect number

- Step 1: Start
- Step 2: Input low, high
- Step 3: Repeat for n = low to high
- Step 4: C = 0
- Step 5: Repeat for i = 1 to n
- Step 6: If n ≡ 0 (mod i) then C ← C+i
- Step 7: End if of Step 6
- Step 8: End of loop of Step 5
- Step 9: If C= n*2 then Print C
- Step 10: End if of Step 9
- Step 11: End of loop of Step 3
- Step 12: Stop

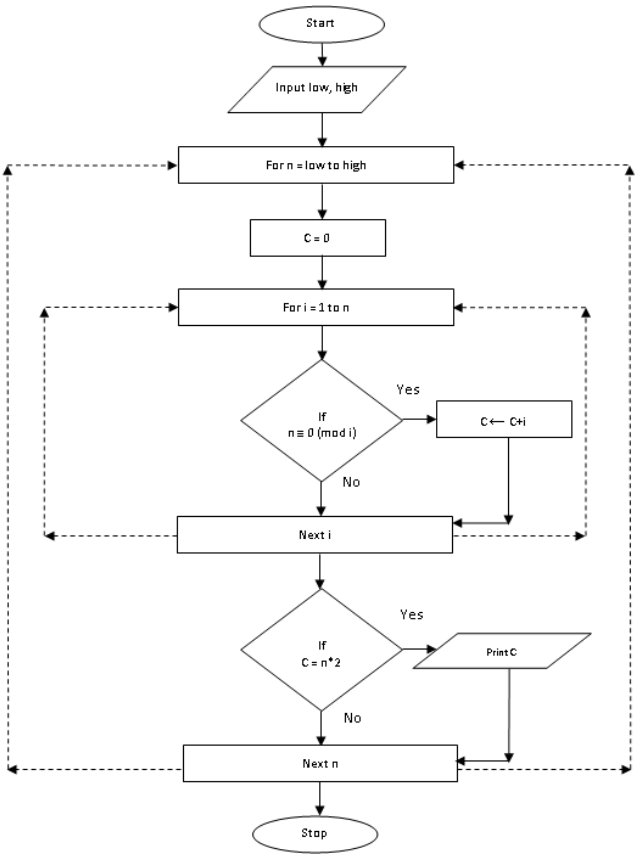

Fig. 6. Flow Chart for Perfect number

8. Algorithm for twin primes

- Step 1: Start
- Step 2: C = 0
- Step 3: Input low, high
- Step 4: Repeat for i = low to high-2
- Step 5: res1 = isPrime(i)
- Step 6: res2 = isPrime(i+2)
- Step 7: If res1 = True and res2 = True then
- Step 8: Print i, i+2
- Step 9: C ← C+1
- Step 10: End if of Step 7
- Step 11: End of loop of Step 4
- Step 12: Print C
- Step 13: Stop
- Step 14: Function isPrime(n)
- Step 15: If n = 1 then return False
- Step 16: Else
- Step 17: Repeat for i =2 to i = n/2
- Step 18: If n ≡ 0 (mod i) then return False
- Step 19: End if of Step 18
- Step 20: End of loop of Step 17
- Step 21: Return True
- Step 22: End of if of Step 15
- Step 23: End of function

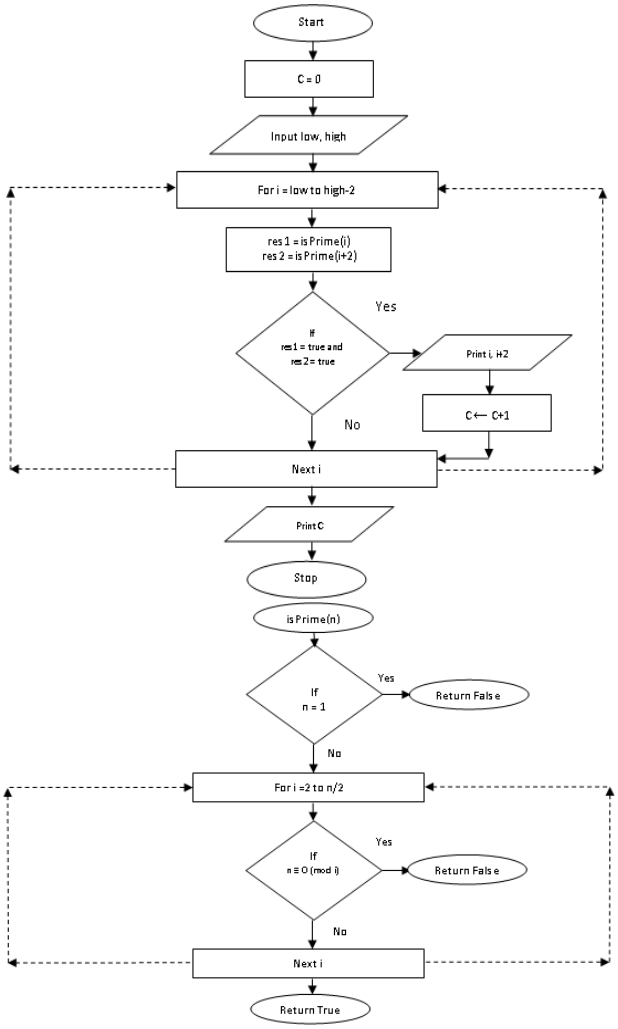

Fig. 7. Flowchart for twin primes

9. Algorithm for geometrical Pythagorean triples

- Step 1: Start
- Step 2: Input low, high
- Step 3: Repeat for a = low to high

- Step 4: Repeat for b = low to high
- Step 5: Repeat for c = low to high
- Step 6: If $a^2 + b^2 = c^2$ then
- Step 7: Print a, b, c
- Step 8: End if of Step 6
- Step 9: End of loop of Step 5
- Step 10: End of loop of Step 4
- Step 11: End of loop of Step 3
- Step 12: Stop

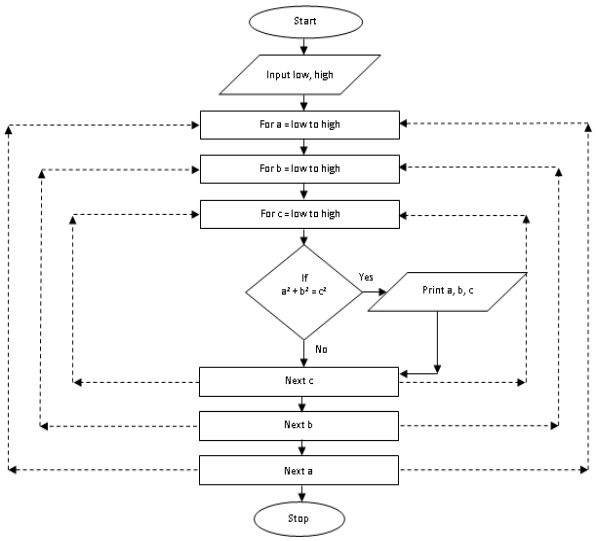

Fig. 8. Flow chart for geometrical Pythagorean triples

10. Conclusion

We have generated algorithms and flow charts for number theoretic functions which are well defined in a given range. These algorithms are useful to find the values of number theoretic functions within the specific interval. These algorithms can be used for designing the cryptographic schemes based on number theoretic functions and special numbers.

References

[1] Apostol, Tom M., "Introduction to Analytic Number Theory", Springer Undergraduate Texts in Mathematics, (1976).
[2] Apostol, Tom M., "Modular Functions and Dirichlet Series in Number Theory (2nd Edition)", New York: Springer, (1989).
[3] Cohen, H., "A Course in Computational Algebraic Number Theory", Berlin: Springer, (1993).
[4] Burton, D. M., "Elementary Number Theory", Tata McGraw-Hill Publ. Comp. Ltd., New Delhi, (2006).
[5] Andrews, G. E., "Number Theory", Hindustan Publ. Corp. Delhi, (1992).
[6] Agarwal, S. & Uniyal, A.S., "Prime Weighted Graph in Cryptographic System for Secure Communication", International Journal of Pure and Applied Mathematics (IJPAM), Volume 105, No. 3, (2015), page: 325-338.
[7] Agarwal, S. & Uniyal, A.S., "Formation of Abelian Group by Mobius (α,β) Function", International Journal of Science and Research (IJSR), (ISSN: 2319-7064), Volume 4, Issue 10, October (2015), page: 177-179.
[8] Agarwal, S. & Uniyal, A.S., "Enhancing the Security of ATM Password using Multi-dimensional Tree", International Journal of Mathematics Research (IJMR), Volume 9, No. 1, (2017), page: 53-58.
[9] Agarwal, S. and Uniyal, A.S., "Number Theoretic Functions: Augmentation and Analytical Results", International Journal of Mathematics Trends and Technology (IJMTT), Volume 43, No. 3, March (2017), page: 232-233.
[10] Radha K., Balaji G. & Jaya Sudha VP, "Number Theory in Providing Security", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July (2013), page: 254-256.
[11] H. W. Lenstra, JR., "Algorithms in Algebraic Number Theory", Bulletin (New Series) of The American Mathematical Society Volume 26, Number 2, April, (1992), page: 211-244.