

Assignments for C Programming Language

Dr Avishek Adhikari
Department of Mathematics,
Presidency University, Kolkata

1. Write a C program that can swap two integers.
2. Using exactly two variables, write a C program that can swap two integers.
3. Given the lengths of the three straight lines, write a C program that can check whether
 - (a) the three straight lines form a triangle or not.
 - (b) the three straight lines form an equilateral triangle or not.
 - (c) the three straight lines form an isosceles triangle or not.
4. Given the co-ordinates for the three points A, B and C as (a_1, b_1) , (a_2, b_2) and $(a_3, b_3) \in \mathbb{R}^2$, respectively, write a C program that can
 - (a) find *area* of the triangle ABC.
 - (b) check whether the triangle ABC is a right angle triangle or not.
5. Given n integers, write a C program that can find *exact average* (in float) and *integer average* of the n numbers.
6. For given n , write a C program that can print
$$\begin{array}{ccccccc} & & & & 1 & & \\ & & & & 1 & 2 & \\ & & & & & \dots & \\ & & & & 1 & 2 & \dots & n \end{array}$$
7. Given n , write a C program that can compute factorial of n .
8. Write a C program that can find the value of nP_r .
9. Write a C program that can find the value of nC_r .
10. Write a C program that can find the *gcd* of two integers.

11. Write a C program that can find *lcm* of two integers.
12. Given n write a C program that can print first n terms of *Fibonacci series*. Also compute the sum of the series.
13. Given n , write a C program that can find the *pascal triangle*.
14. Write a C program that can check whether a given integer is *prime* or not.
15. Given n , write a C program that can compute the sum $1 + (1 + 2) + \cdots + (1 + 2 + \cdots + n)$. Also print $1 + (1 + 2) + \cdots + (1 + 2 + \cdots + n) = \text{sum}$
16. Given n , write a C program that can compute $1^2 + 2^2 + \cdots + n^2$.
17. Given n and k , write a C program that can compute $1^k + 2^k + \cdots + n^k$.
18. Given n , write a C program that can find all positive factors of n .
19. Given n , write a C program that can find all prime factors of n .
20. Given $n > 1$, write a C program that can represent n as $p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, where p_i s are distinct primes and α_i s are positive integers, for $i = 1, 2, \dots, k$.
21. Given a positive integer n , write a C program that can find the set $\mathbb{Z}_n^* = \{x \in \mathbb{Z}_n : \gcd(x, n) = 1\}$.
22. Given a positive integer n , write a C program that can find all units of the ring $(\mathbb{Z}_n, +, \cdot)$.
23. Given a positive integer n and an element $(a) \in \mathbb{Z}_n$, write a C program that can find (b) in \mathbb{Z}_n such that $(a)(b) = (1)$, if it exists.
24. Given n , write a C program that can find $\phi(n)$.
25. Given n , write a C program that can find $\tau(n)$.
26. Given n , write a C program that can find $\sigma(n)$.
27. Given n integers, $n < 100$, using C programming, write down the integers in ascending order.
28. Given marks of n subjects for m students, write a C program that can find *average* mark of each student and find *average* mark of each subject.
29. Write a C program that can compute the *addition* of two matrices, report if not possible.

30. Write a C program that can compute the *multiplication* of two matrices, report if not possible.
31. Write a C program that can compute the *transpose* of a given matrix.
32. Write a C program that can check whether a given matrix is *symmetric* or not.
33. Write a C program that can check whether a given matrix is *skew-symmetric* or not.
34. Write a C program that can demonstrate that every square matrix (over \mathbb{R}) can be written as *sum of symmetric matrix and skew-symmetric matrix*.
35. Write a C program that can find the *trace* of a given matrix.
36. Write a C program that can find the *sum of cross-diagonal elements* of a given matrix.
37. Write a C program that can find the *inverse* of a given matrix, if it exists.
38. Given an 3-digit integer, write a C program that can find the reverse integer, i.e., if the integer is abc, then the program will output cba.
39. Given a positive integer n , write a C program that can construct a Latin square of order n .
40. Given an odd positive integer n , write a C program that can construct a Latin square of order n in which all the diagonal elements are distinct.
41. Given an odd positive integer n , write a C program that can construct a Latin square of order n in which all the diagonal elements are in ascending order.
42. Write a C program that can transform a decimal positive integer into binary representation.
43. Write a C program that can transform a decimal positive integer into octal representation.
44. Write a C program that can transform a binary representation into decimal representation.
45. Given a positive integer n , write a C program that can print all subsets of the set $\{1, 2, \dots, n\}$.
46. Given a positive integer n , write a C program that can print all subsets of odd order of the set $\{1, 2, \dots, n\}$.

47. Given a positive integer n , write a C program that can print all subsets of even order of the set $\{1, 2, \dots, n\}$.
48. Given the Boolean coefficients a_1, a_2, a_3 of the Boolean function $f(x_1, x_2, x_3) = a_1x_2 + a_2x_2 + a_3x_3$ in three variables x_1, x_2, x_3 , write the truth table for the function f .
49. Given an interval $[a, b]$, write a C program that can find all twin primes within the interval $[a, b]$. If there exists no such twin primes, report that there exists no twin primes in the interval $[a, b]$.
50. Given an interval $[a, b]$, write a C program that can find all perfect numbers within the interval $[a, b]$. If there exists no such perfect number, report that there exists no perfect number in the interval $[a, b]$.
51. Given n , write a C program (using function) that can compute factorial of n .
52. Given an interval $[a, b]$, write a C program using function that can find all primes within the interval $[a, b]$. If there exists no such prime, report that there exists no prime in the interval $[a, b]$.
53. Write a C program that justifies prime number theorem.
54. Write a C program (using function) that can find the value of nP_r .
55. Write a C program (using function) that can find the value of nC_r .
56. Write a C program (using function) that can find the gcd of two integers.
57. Write a C program (using function) that can find lcm of two integers.
58. Write a C program (using function and pointer) that can swap two integers.
59. Write a C program (using function) that can check whether a given integer is *prime* or not.
60. Given n , write a C program (using gcd function) that can find $\phi(n)$. find $\sigma(n)$.
61. Write a C program (using function) that can transform a decimal positive integer into binary representation.
62. Write a C program (using function) that can transform a decimal positive integer into octal representation.
63. Write a C program (using function) that can transform a binary representation into decimal representation.

64. Write a C program that can count the number of characters in a given string of input.
65. Write a C program that can count the number of consonants and vowels in a given string of input.
66. Write a C program that is capable of doing name abbreviation.
67. Write a C program that can check whether a given string of input is palindrome or not.
68. Write a C program that can change the lower case alphabet into upper case alphabet.
69. Write a C program that can change the upper case alphabet into lower case alphabet.
70. Write a C program that can change the all lower case alphabet into upper case alphabet and all upper case alphabet into lower case alphabet in a given string of alphabets.
71. Write a C program that takes a sentence as input and outputs a sentence from backward direction. For example, input: “I am a good boy”; output: “boy good a am I”.
72. Given n integers, using dynamic memory allocation write a C program that can output the integers in ascending order.
73. Given marks of n subjects for m students, write a C program (using dynamic memory allocation) that can find *average* mark of each student and find *average* mark of each subject.
74. Write a C program (using dynamic memory allocation) that can compute the *addition* of two matrices, report if not possible.
75. Write a C program (using dynamic memory allocation) that can compute the *multiplication* of two matrices, report if not possible.
76. Write a C program (using dynamic memory allocation) that can compute the *transpose* of a given matrix.
77. Write a C program (using dynamic memory allocation) that can check whether a given matrix is *symmetric* or not.
78. Write a C program (using dynamic memory allocation) that can check whether a given matrix is *skew-symmetric* or not.
79. Write a C program (using dynamic memory allocation) that can demonstrate that every square matrix (over \mathbb{R}) can be written as *sum* of *symmetric matrix* and *skew-symmetric matrix*.

80. Write a C program (using dynamic memory allocation) that can find the *trace* of a given matrix.
81. Write a C program (using dynamic memory allocation) that can find the *sum of cross-diagonal elements* of a given matrix.
82. Write a C program (using dynamic memory allocation) that can find the *inverse* of a given matrix, if it exists.
83. Given a positive integer n , write a C program (using dynamic memory allocation) that can construct a Latin square of order n .
84. Given an odd positive integer n , write a C program (using dynamic memory allocation) that can construct a Latin square of order n in which all the diagonal elements are distinct.
85. Given an odd positive integer n , write a C program (using dynamic memory allocation) that can construct a Latin square of order n in which all the diagonal elements are in ascending order.
86. Write a C program (using function and dynamic memory allocation) in which given n integers will be arranged in ascending order.
87. Write a C program that can print in the terminal the program itself.
88. Write a C program that can find the number of alphabets in a given file stored in the same folder as the program itself.
89. Write a C program that can read an input n (the order of a latin square) from a given file stored in the same folder as the program itself and write the output (latin square of order n) in the same file.
90. Write a C program using break.
91. Write a C program using continue.
92. Write a C program using switch-case.
93. Given n , write a C program (using function) that can compute factorial of n and store the output in a file.
94. Write a C program (using function) that can find the value of nP_r and store the output in a file.

95. Write a C program (using function) that can find the value of nC_r and store the output in a file.
96. Write a C program (using function) that can find the *gcd* of two integers and store the output in a file.
97. Write a C program (using function) that can find *lcm* of two integers and store the output in a file.
98. Write a C program (using function and pointer) that can swap two integers and store the output in a file.
99. Write a C program (using function) that can check whether a given integer is *prime* or not and store the output in a file.
100. Given n , write a C program (using gcd function) that can find $\phi(n)$ find $\sigma(n)$ and store the output in a file.
101. Given n integers, using dynamic memory allocation write a C program that can output the integers in ascending order and store the output in a file.
102. Given marks of n subjects for m students, write a C program (using dynamic memory allocation) that can find *average* mark of each student and find *average* mark of each subject and store the output in a file.
103. Write a C program (using dynamic memory allocation) that can compute the *addition* of two matrices, report if not possible and store the output in a file.
104. Write a C program (using dynamic memory allocation) that can compute the *multiplication* of two matrices, report if not possible and store the output in a file.
105. Write a C program (using dynamic memory allocation) that can compute the *transpose* of a given matrix and store the output in a file.
106. Write a C program (using dynamic memory allocation) that can check whether a given matrix is *symmetric* or not and store the output in a file.
107. Write a C program (using dynamic memory allocation) that can check whether a given matrix is *skew-symmetric* or not and store the output in a file.
108. Write a C program (using dynamic memory allocation) that can demonstrate that every square matrix (over \mathbb{R}) can be written as *sum* of *symmetric matrix* and *skew-symmetric matrix* and store the output in a file.

109. Write a C program (using dynamic memory allocation) that can find the *trace* of a given matrix and store the output in a file.
110. Write a C program (using dynamic memory allocation) that can find the *sum of cross-diagonal elements* of a given matrix and store the output in a file.
111. Write a C program (using dynamic memory allocation) that can find the *inverse* of a given matrix, if it exists and store the output in a file.
112. Given a positive integer n , write a C program (using dynamic memory allocation) that can construct a Latin square of order n and store the output in a file.
113. Given an odd positive integer n , write a C program (using dynamic memory allocation) that can construct a Latin square of order n in which all the diagonal elements are distinct and store the output in a file.
114. Given an odd positive integer n , write a C program (using dynamic memory allocation) that can construct a Latin square of order n in which all the diagonal elements are in ascending order and store the output in a file.
115. Write a C program that can transform a decimal positive integer into binary representation and store the output in a file.
116. Write a C program that can transform a decimal positive integer into octal representation and store the output in a file.
117. Write a C program that can transform a binary representation into decimal representation and store the output in a file.
118. Write a C program (using function and dynamic memory allocation) in which given n integers will be arranged in ascending order and store the output in a file.
119. Write a C program that can count the number of characters in a given string of input and store the output in a file.
120. Write a C program that can count the number of consonants and vowels in a given string of input and store the output in a file.
121. Write a C program that is capable of doing name abbreviation and store the output in a file.
122. Write a C program that can check whether a given string of input is palindrome or not and store the output in a file.

123. Write a C program that can change the lower case alphabet into upper case alphabet and store the output in a file.
124. Write a C program that can change the upper case alphabet into lower case alphabet and store the output in a file.
125. Write a C program that can change the all lower case alphabet into upper case alphabet and all upper case alphabet into lower case alphabet in a given string of alphabets and store the output in a file.
126. Write a C program that can print in the terminal the program itself and also store the output in a file.
127. Write a C program that can find the number of alphabets in a given file stored in the same folder as the program itself and store the output in a file.
128. Write a C program using break and store the output in a file.
129. Write a C program using continue and store the output in a file.
130. Write a C program using switch-case and store the output in a file.