

Starts: 19/11/2025 at 2:30 PM IST (Duration: 1 hour 45 mins)

Total marks: 25

Instructions

- All programs should be compatible with GNU C.
- Late submissions will attract marks deduction.
- You are advised to compile the code before submitting it to WeLearn. If a code does not compile at our end, marks will be deducted for the problem.
- Each program should follow a strict naming convention: **QNo.c** (e.g. Q1.c, Q2a.c etc.). Programs not adhering to the convention will not be evaluated.
- All codes (.c files) should be submitted in a single zipped folder to WeLearn.
- You should put appropriate comments in the code.

1. (**Marks: 12**) Given a sorted (ascending) array (length n) of positive integers (e.g., $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$), write a program to check if a given sum (e.g., 19) exists between two elements (here, between 9 and 10) in $\mathcal{O}(n)$ time.
2. (**Marks: 5**) Write a **recursive** function **SqSum()** to compute the *squared sum* of the first n integers (i.e., $1^2 + 2^2 + \dots + n^2$) and print the same, where n is given as *input* by the user and passed as an argument to **SqSum()**.
3. (**Marks: 8**) Given two complex numbers (stored as arrays; real part as element 0, imaginary part as element 1), write your own function **addComplex()** to take these arrays as arguments and return the sum (sums of the real and imaginary parts separately), store in your own header file **myheader.h** and call the same from main. Both the real and imaginary parts are integers.