

C++: Practical session 4

1 Functions and references

Write and test the following functions:

1. A function **sum** that takes two integers and returns their sum.
2. A function **decrease** that takes two integers **a** and **b** and reduces **a** by the value of **b**, and does not return anything. However, the variable passed as **a** should be updated so that its decreased value is available to the calling function.
3. A function **q** that takes three **doubles** a, b, c , and another two $x1, x2$ by reference, and returns in $x1$ and $x2$ any solutions of the quadratic $ax^2 + bx + c = 0$. The return value of the function should be the number of solutions found (an integer). Ignore complex numbers for this case.

2 Arrays

2.1 1D Array

Write a function that computes partial sums of an array, and puts them into a separate array. The function should take two arrays of length 20 as arguments, one as input and the other for output.

The partial sums for an array a_i are $b_k = \sum_{i=0}^k a_i$, i.e. the first partial sum of **a** is **a[0]**; the second partial sum is **a[0] + a[1]**, the third is **a[0] + a[1] + a[2]**, etc.

Use this function to print compute the first 20 triangular numbers, using $a_i = i$.

2.2 Matrix operations

Write a program that takes a 3-by-3 matrix as input from the user, and outputs its determinant and its trace. The matrix should be stored as a nested array, and the determinant and trace routines should be implemented as separate functions. You should also make the array passed to the functions **const**.

The awkwardness of writing this should make you realise that C++ is not good at handling matrices.