Practical 10

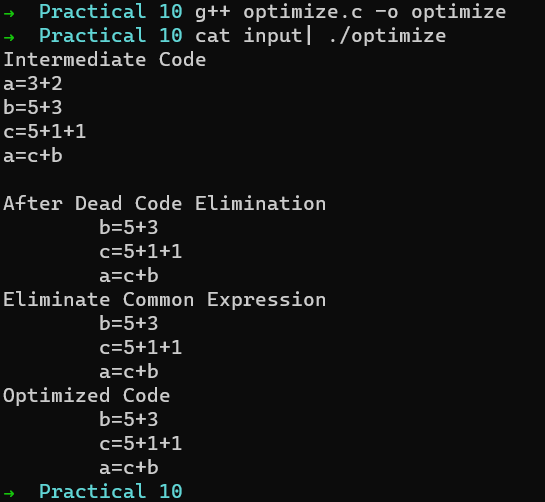
Name: Tushar Harsora

Roll No: 19BCE509

# Introduction

This practical is focused on the last remaining component in compiler construction. We learn about various code optimization techniques. We explored and studied various way to speed up the execution of the source code and techniques to make source compact. We are demonstrating Dead code elimination, common subexpression elimination. There are also advanced ways to improve the execution of program like placing recently used data in cache, using wide registers and instructions like SEE, AVX for faster vector and matrix processing. The techniques like loop unrolling, align jumps for 32 bit or 64 bit boundary, use fast math with expense of less precision, introduce loop invariants and reduce computations, peephole optimizations etc.

# Screenshots



Above screenshot has 2 passes first one removes dead code and second removes common sub expression. The final output is optimized code.

# Conclusion

In conclusion we learnt new and improved techniques for machine code optimization. Some of the optimization techniques also works on higher level source code. We also studied how optimization passes are implemented in compilers like LLVM, GCC, and just in time compilers for languages like JAVA, .NET, etc.