

Lab 1

Exercise 1.1.4. A compiler that translates a high level language into another high level language is called a **source to source translator**. What advantages are there to using C as a target language for a compiler?

Answer.

The C language is very versatile, by using C as the target language for the source to source translator one can ensure that the code can be run on a number of systems. This is due to the fact that for the C language there are many compilers available, on a plethora of different systems.

Exercise 1.6.1. For the block-structured C code below, indicate the values assigned to w, x, y , and z .

```
int w, x, y, z;
int i = 4; int j = 5;
{
    int j = 7;
    i = 6;
    w = i + j;
}
x = i + j;
{
    int i = 8;
    y = i + j;
}
z = i + j;
```

Solution.

$w = 13, x = 11, y = 13, z = 11$

Exercise 1.11. The **Measure Of Software Similarity** (MOSS) [SWA03] tool can detect similarity of programs written in a variety of modern programming languages. Its main application has been in detecting similarity of programs submitted in computer science classes, where such similarity may indicate plagiarism (students, beware!). In theory, detecting equivalence of two programs is **undecidable**, but MOSS does a very good job of finding similarity in spite of that limitation.

Investigate the techniques MOSS uses to find similarity. How does MOSS differ from other approaches for detecting possible plagiarism?

Answer.

When someone changes the variable name or attempts to add additional white space or otherwise changing minute details within the formatting of a program, in an attempt to deceive or cover the fact that a piece of code has been copied from elsewhere, it might be enough to fool the human eye. However, MOSS does more than simply directly compare the code that is processed through it. While those changes to alter the appearance of the code, the overall structure of the program remains unchanged, the number of token and line matches between the documents also remain the same. An example of MOSS output can be found below.

Moss Results				
Sun Mar 14 15:24:02 PST 1999				
Options -l c -m 10				
[Text Report] [How to Read the Results] [Tips] [FAQ] [Contact Moss] [Submission Scripts] [Credits]				
File 1	File 2	Tokens Matched	Lines Matched	
mike_wolf.c (79%)	mike_fox.c (80%)	463	139	
bill_smyth.c (86%)	bill_smith.c (88%)	456	133	
jane_white.c (59%)	jane_blanco.c (68%)	354	111	
john_doe.c (100%)	john_deer.c (100%)	220	49	
Any errors encountered during this query are listed below.				

The threshold of what is considered "plagiarism" is to be determined by the professor or whoever is in charge of making such judgement.

Although MOSS is quite efficient compared to other systems that simply rely on direct comparison of text, it can be outsmarted with subtle changes in the logic of the code.

```
var A = 10;

if(A>5)
  console.log("Greater than 5...");
else
  console.log("Less than or equal to 5...");

/*+++++++ VS ++++++*/

if(A<=5)
  console.log("Less than or equal to 5...");
else
  console.log("Greater than 5...");
```

Exercise 3.1. Assume the following text is presented to a C scanner:

```

main(){
    const float payment = 384.00;
    float bal;
    int month = 0;
    bal=15000;
    while (bal>0){
        printf("Month: %2d  Balance: %10.2f\n", month, bal);
        bal=bal-payment+0.015*bal;
        month=month+1;
    }
}

```

What token sequence is produced? For which tokens must extra information be returned in addition to the token code?

Solution.

$\langle id, "main" \rangle \langle openParen \rangle \langle closeParen \rangle \langle openBrace \rangle \langle const \rangle \langle float \rangle \langle id, "payment" \rangle \langle assign \rangle \langle literal, "384.00" \rangle$
 $\langle semiColon \rangle \langle float \rangle \langle id, "bal" \rangle \langle semiColon \rangle \langle int \rangle \langle id, "month" \rangle \langle assign \rangle \langle literal, "0" \rangle \langle semiColon \rangle \langle id, "bal" \rangle$
 $\langle assign \rangle \langle literal, "15000" \rangle \langle semiColon \rangle \langle while \rangle \langle openParen \rangle \langle id, "bal" \rangle \langle greaterThan \rangle \langle literal, "0" \rangle \langle closeParen \rangle$
 $\langle openBrace \rangle \langle printf \rangle \langle openParen \rangle \langle literal, "Month : %2d Balance : %10.2f\n" \rangle \langle id, "month" \rangle \langle id, "bal" \rangle$
 $\langle semiColon \rangle \langle id, "bal" \rangle \langle assign \rangle \langle id, "bal" \rangle \langle minus \rangle \langle id, "payment" \rangle \langle plus \rangle \langle literal, "0.015" \rangle \langle multiply \rangle \langle id, "bal" \rangle$
 $\langle semiColon \rangle \langle id, "month" \rangle \langle assign \rangle \langle id, "month" \rangle \langle plus \rangle \langle literal, "1" \rangle \langle semiColon \rangle \langle closeBrace \rangle \langle closeBrace \rangle$
