

Örebro University
School of Science and Technology
[Thomas Padron-McCarthy \(thomas.padron-mccarthy@oru.se\)](mailto:thomas.padron-mccarthy@oru.se)

Exam

Compilers and Interpreters

for Dataingenjörsprogrammet, and others

Saturday October 30, 2010

Exam for:
DT3004 Datateknik C, Kompilatorer och interpretatorer, provkod 0100

→ This exam is also available in a Swedish version.

Aids:	No aids.
Score	Maximum score is 32.
requirements:	To pass (3 or G), at least 16 points are required.
Results:	Announced on the course website or by e-mail by Saturday November 20, 2010.
Return of the exams:	After the result has been announced, exams can be collected from the university's central "tentamensutlämning".
Examiner:	Thomas Padron-McCarthy

- Write clearly. Solutions that can not be read can obviously not give any points. Unclear and ambiguous wording will be misinterpreted.
- Enter the personal exam code on each sheet submitted. Do not write your name

on the sheets.

- Write only on one side of the paper. Do not use red letters.
 - Assumptions beyond those in the given problems must be stated.
 - You are allowed to explain your solutions. Even an incorrect answer may give some points, if the key ideas were right.
-

GOOD LUCK!!

Task 1: Phases (3 p)

When we compile the following C program, the compiler gives the italicized error and warning messages:

```
#include <stdio.h>

int main(void) {
    int a;

    printf("Hi!\n ");           error: missing terminating " character
    printf("Give a number: ");
    scanf("%d", &a ;          error: expected ')' before ';' token
    printf("The number was: %d\n", a);

    return "Charles";         warning: return makes integer from pointer without a cast
}
```

A compiler's work is usually divided into several phases. In which phases are these errors and warnings detected?

Task 2: Scanning and Regular Expression (5 p)

a) (2p) Write regular expressions for the following:

- a year with four digits (that is, from year 1000 to year 9999)
- clothes sizes (which can be XS, S, M, L and XL)

b) (2p) Write a regular expression for the Swedish personal identity number (for example **631211-1658**). Your expression should match all valid such numbers. It is difficult to write a regular expression that does not also match certain incorrect numbers, so your solution is allowed to do that. But state at least one check that is *not* made by your regular expression.

c) (1p) What is the difference between a token and a lexeme?

Task 3: Grammars (10 p)

Here are three things that are problematic in a grammar:

- left recursion
- FIRST() conflicts
- ambiguity

For each of these problems, provide an example of a grammar that exhibits the problem. Also explain, for each of the grammars, how the problem manifests itself in practice. (That is: what is it that does not work, because of the problem?) Also explain how to solve the problem.

Task 4: Intermediate Code (5 p)

```

x = 1;
y = 2;
z = 3;
while (y == 2) {
    if (z > 4) {
        y = y - 1 - 1;
    }
    else {
        z = z + y * z + z;
        t = t + 2;
    }
}

```

Translate the above program section to *two* of the following three types of intermediate code:

- an abstract syntax tree (by drawing the tree!)
- postfix code for a stack machine
- three-address code

Note: There are three sub-tasks in the task above. Select and answer (at most) *two* of them. (If you answer all three, the one with the most points will be discarded.)

Task 5: Some Terms(9 p)

Briefly explain what the following terms from compiler technology mean:

- target language
 - target program
 - front end
 - Yacc
 - symbol table
 - shift-reduce conflict
 - deterministic finite state machine
 - reserved word (or "keyword")
 - call sequence
-