

Ministry of Higher Education and Scientific Research

**Salahaddin University / Erbil
College of Engineering
Dept. of Software Engineering**

**2nd Term Exam
2011-2012**

**Subject: Compilers
Time: 90 Minutes
Lecturer: Amanj Sherwany
Date: 6 May 2012**

The highest obtainable mark is 100, the minimum passing mark is 50

Q1: (16 points)

For each of the following data structures, show how their instances are represented in the memory of a 32-bit machine, state their sizes in bytes, and the possible needed alignments:

- `struct element{char x; int y; double z;}`
- `union element{char x; int y; double z;}`

* * *

Q2: (30 points)

Translate the following function to assembly:

```
int factorial(int n){
    if(n == 0)
        return 1;
    if(n == 1)
        return 1;
    return (n * factorial(n-1));
}
```

The target machine is a 32-bit RISC with 4-byte integers, a stack pointer register SP, a return address register RA, eight general purpose registers R0 to R7, and the following instruction set:

<code>goto label</code>	
<code>if reg < opnd goto label</code>	(==, >=, etc)
<code>move dst, opnd</code>	assign opnd to dst
<code>add dst, reg1, opnd2</code>	assign reg1 plus opnd2 to dst ditto for sub, mul, and, or, leftshift, etc
<code>load dst, (reg1 + opnd2)</code>	read integer from memory at reg1+opnd2
<code>store (reg1 + opnd1), opnd3</code>	write integer (opnd3) to memory at reg1+opnd2
<code>call label</code>	set RA to next instruction then jump to label
<code>return</code>	jump to the address in RA

Each *dst* must be a register, and each *opnd* must be a register or an integer constant. The function call

conventions are that parameters are passed in register *R0*, the return value is returned in *R1*, and a function call may destroy any general-purpose register and *RA*. The stack grows from high to low addresses, and *SP* should always point to the lowest word of the current stack frame.

Apart from the statements and expressions, include the code for setting up the stack frame, storing registers in the stack frame, and fetching registers from the stack frame.

* * *

Q3: (30 points)

Translate the following C-code to RTL/quadruples:

```
int multiply(int x, int y){
    int result = 0;
    for(int i = y; i > 0; i--){
        result += x;
    }
    return result;
}
```

* * *

Q 4: (24 points)

For the following intermediate code representation:

- Find the basic blocks (**6 points**)
- Draw the Control-Flow Graph (CFG) (**4 points**)
- Calculate the Liveness (LIVEIN and LIVEOUT) for each instruction (**10 points**)
- Calculate the Liveness (LIVEIN and LIVEOUT) for each basic block (**4 points**)

```
I1.    a := b + 3
I2.    if a == 10 goto L1
I3.    f := a - b
I4.    goto L2
I5. L1: f := a + b
I6. L2: return f
```

Good Luck