

CSc 372, Fall 1996

Final Examination Solutions

Problem 1 (12 points):

For each of ML, C++, Icon, and Prolog, cite three elements in the language that are unique to that language among this group of four.

ML: Type deduction, partial application of functions, no variables

C++: Supports an object oriented paradigm, type extensibility, templated functions.

Icon: Expression failure, string scanning, a compact but expressive set of built-in functions.

Prolog: Procedures (predicates) with multi-faceted behavior, the notion of unification, expressions represented as trees that are evaluated on demand.

Problem 2 (8 points):

Select one element from EACH set of three in the previous problem and describe a benefit it provides to the programmer.

ML: A partial application can be used to create a more specialized function from a more general function.

C++: Encapsulation of data in objects allows the data to change without requiring modification to code that uses those objects.

Icon: The set of built-in functions is compact enough that I'm personally able to write Icon programs of a significant size without needing to consult any reference material.

Prolog: The multi-faceted behavior of predicates allows one predicate to serve several purposes. For example, `member/2` can be used to test for membership, generate elements, and more.

Problem 3 (5 points):

Write a predicate `permute(L, P)` that for a list `L` of 1, 2, or 3 elements instantiates `P` to each permutation of the elements of `L`.

```
permute([A], [A]).
permute([A, B], [A, B]).
permute([A, B], [B, A]).
permute([A, B, C], [A, B, C]).
permute([A, B, C], [A, C, B]).
permute([A, B, C], [B, A, C]).
permute([A, B, C], [B, C, A]).
permute([A, B, C], [C, A, B]).
permute([A, B, C], [C, B, A]).
```

Problem 4 (3 points):

The instructor's implementation of `roman/2` used a series of facts of this form:

```
rdval('I',1).
rdval('V',5).
rdval('X',10).
etc.
```

Consider an attempt at an ML function to provide the same functionality as `rdval/2`:

```
fun rdval("I") = 1
  | rdval("V") = 5
  | rdval("X") = 10
  etc.
```

Without particular regard to usage in `roman/2`, is the ML function a good approximation of the `rdval/2` predicate?

The ML function provides only half the functionality of the predicate, which can also map from given decimal numbers to their Roman equivalent.

Problem 5 (7 points):

Write a predicate `sum_ints(L, Sum)` that produces a sum of the integers in list `L`. `L` might contain things other than integers, but they should be ignored.

```
sum_ints([], 0).
sum_ints([H|T], Sum) :- integer(H), sum_ints(T, TSum),
                        Sum is H+TSum, !.
sum_ints([_|T], Sum) :- sum_ints(T, TSum), Sum is TSum.
```

Problem 6 (6 points):

Write a predicate `assemble(L, Segments)` that describes the relationship that the list `L` can be assembled from two of the lists in `Segments`.

```
assemble(L, Segments) :- member(A, Segments), member(B, Segments),
                          append(A, B, L).
```

Problem 7 (10 points):

```
cfa([], _, []).
cfa([Amt|Amts], Coins, [Change|MoreChange]) :-
  make_change(Coins, Amt, Change, Left),
  cfa(Amts, Left, MoreChange).
```

Problem 8 (8 points):

Write an Icon procedure `extract(s, m, n)` that extracts a portion of a string that represents a hierarchical data structure. `m` is a major index and `n` is a minor index. Major sections of the string are delimited by slashes and are composed of minor sections separated by colons. Here is a sample string:

```
/a:b/apple:orange/10:2:4/xyz/
```

It has four major sections which in turn have two, two, three and one minor sections.

```
procedure extract(s, i, j)
    return split(split(s, '/') [i], ':') [j]
end
```

Problem 9 (6 points)

Write an Icon program that reads, on standard input, a list of words, one per line, and prints the words that contain the letters a, b, and c in that order. The letters need not be consecutive and there may be more than one occurrence of each.

```
procedure main()
    while line := read() do
        line ? {
            tab(upto('a')) & tab(upto('b')) & tab(upto('c')) &
            write(line)
        }
end
```

Problem 10 (6 points)

Write an Icon procedure `revby2(s)` that reverses the string on a character pair-wise basis and returns the resulting string. **NOTE: Your solution must use string scanning. In particular you may not use string subscripting, sectioning, or the * operator.**

```
procedure revby2(s)
    r := ""

    s ? {
        tab(0)
        while r || := move(-2)
            pos(1) | fail
    }

    return r
end
```

Problem 11 (7 points):

Write an Icon program that prints on standard output, one per line, each minute of the day in the form 12:22pm.

```
procedure main()
  every f("am")
  every f("pm")
end

procedure f(m)
  every write(right(12 | (1 to 11), 2, " "), ":",
    right(0 to 59, 2, "0"), m)
end
```

More concisely:

```
procedure main()
  every m := ("am"|"pm") &
    write(right(12 | (1 to 11), 2, " "), ":",
      right(0 to 59, 2, "0"), m)
end
```

Problem 12 (4 points):

Write an Icon program `rev` that reads a file redirected to standard input and writes out the lines in the file in reverse order—last line first, first line last.

```
procedure main()
  L := []
  while push(L, read())
  while write(pop(L))
end
```

Problem 13 (9 points):

Write an ML function `samesums(L)` of type `(int * int * int) list -> bool` that tests whether all the 3-tuples in `L` have the same sum.

```
fun sum3(a,b,c) = a+b+c:int

fun samesums([]) = true
  | samesums([t]) = true
  | samesums(t1::t2::ts) =
    sum3(t1) = sum3(t2) andalso samesums(t2::ts);
```

Problem 14 (9 points)

Write code for a C++ class named X that exhibits the following elements...

```
#include <iostream.h>

class X {
public:
    X(int a, int b) : itsSum(a+b) {}
    int f() { return itsSum; }
    void make5();
private:
    X() :itsSum(0) {}
    int itsSum;
};

void X::make5()
{
    X a, xs[3];
    X *p = new X();

    cout << a.f() + p->f() + xs[0].f() + xs[1].f() + xs[2].f()
         << endl;
}

ostream& operator<<(ostream& o, X& x)
{
    o << "an X at " << &x;
    return o;
}
```

EC 1 (5 points):

Name five programming languages that originated before 1985.

ML, C++, Icon, Prolog, and C.

EC 2 (5 points max):

For one point each, name a programming language and the person generally credited as being the designer of the language.

ML—Robin Milner et al.
C++—Barne Stroustrup
Icon—Ralph Griswold
Prolog—Alain Colmeraurer
Java—James Gosling

EC 3 (1 point):

What is the instructor's favorite programming language?

I guess that if I've got a favorite it would have to be Icon, but it really depends on the

situation. Any answer here was worth a point.

EC 4 (1 point):

Name a popular operating system in which Prolog plays a role in system configuration.

Windows NT

EC 5 (2 points)

Languages can be grouped according to various aspects of the language. Group the languages we studied according to their type checking philosophy.

One possible grouping is that type-checking is done at compile time in ML and C++, but at run time in Icon and Prolog.

EC 6 (3 points)

Write an Icon program that reads a list of words like that described in problem 12 and prints out the words that consist of solely of the hex digits a-f. Examples of such words: added, beef, dead, facade.

```
procedure main()
  while line := map(read()) do
    if *(line -- 'abcdef') = 0 then
      write(line)
  end
```

EC 7 (1 point)

Among ML, C++, Icon, and Prolog, which is your favorite?

Any answer here was worth a point.

EC 8 (2 points)

Of all that we covered, which one language feature did you find most interesting?

For me, it's probably ML's ability to support partial applications of functions.

EC 9 (5 points)

In the same style as problems 1 and 2, name three differences between Java and C++ and for any one of those differences explain the benefit provided to the programmer.

Automatic memory management, function bodies must appear in the class definition, and there can be no global functions or data.

EC 10 (2 points)

Why are static class members an essential element of Java?

Because there are no global functions or global data.

EC 11 (1 point)

What is the Prolog 1000?

A compilation of applications implemented in Prolog.