

# CSc 372 — Comparative Programming Languages

## 37 : Icon — Examples

Christian Collberg  
Department of Computer Science  
University of Arizona  
collberg+372@gmail.com

Copyright © 2005 Christian Collberg

November 30, 2005

### 1 Example 1 (a): Soundex

- When names are communicated by telephone, they are often transcribed incorrectly.
- Soundex is a system of encoding a name that will mitigate the effects of transcription errors.

```
# Convert all occurrences of A,E,H,I,O,  
# U,W,Y in other positions to "."  
# Assign the following numbers to the  
# remaining letters after the first:
```

### 2 Example 1 (a): Soundex

```
# B,F,P,V => 1           L => 4  
# C,G,J,K,Q,S,X,Z => 2   M,N => 5  
# D,T => 3               R => 6
```

```
procedure soundex(name)  
  local first, c, i  
  # Convert to uppercase.  
  name := map(name, string(&lc),string(&uc))  
  
  # Retain the first letter of the name  
  first := name[1]  
  name := map(name, "ABCDEFGHJKLMNOPQRSTUVWXYZ",  
              ".123.12..22455.12623.1.2.2")
```

### 3 Example 1: Soundex

```
# If two or more letters with the same  
# code were adjacent in the original name,  
# omit all but the first
```

```

every c := !"123456" do
    while i := find(c||c,name) do
        name[i+:2] := c
    name[1] := first

    # Now delete our place holder ('.')
    while i := upto('.',name) do name[i] := ""
    return left(name,4,"0")
end

```

## 4 Example 1: Soundex...

`left(s1, i, s2)` shift s1 to the left, append s2:s until position i is reached.

### Example

`COLLBERG`  $\Rightarrow$ (code) "2.441.62"  $\Rightarrow$ (remove duplicates) "2.41.62"  $\Rightarrow$ (restore first) "C.41.62"  $\Rightarrow$ (delete ".")  
"C4162"

`COLBERG`  $\Rightarrow$ (code) "2.41.62"  $\Rightarrow$ (remove duplicates) "2.41.62"  $\Rightarrow$ (restore first) "C.41.62"  $\Rightarrow$ (delete ".")  
"C4162"

## 5 Example 2: Crypt

```

procedure main(args)
    if *args = 1 then
        ky := get(args)
    else {con := open("/dev/tty", "b")
        writes(con, "Enter password: ")
        ky := read(con)
        close(con)
    }
    i := 1; l := 0; k := []
    every put(k, ord(!ky)) do l += 1
    while writes(char(ixor(ord(reads()), k[i]))) do
        i %:= l + 1
end

```

## 6 Example 3: Pack

```

# This programs reads a list of file names from
# standard input and packages the files into a
# single file which is written to standard output.
procedure main()
    while name := read() do {
        close(\in)
        in := open(name) |
            stop("cannot open input file: ", name)
        write("#####")
        write(name)
    }
end

```

```

    while write(read(in))
  }
end

```

## 7 Example 4: Table

```

# Tabulate characters and list each character and
# the number of times it occurs.
# -a Write the summary in alphabetical order of
# the characters. This is the default.
# -n Write the summary in numerical order
# -u Write the characters that occur just once.
link options
procedure main(args)
  local ccount, unique, order, s, a
  local pair, rwidth, opts

  unique := 0 # switch to list unique usage only
  order := 3 # alphabetical ordering switch

```

## 8 Example 4 (b): Table...

```

opts := options(args,"anu")
if \opts["a"] then order := 3
if \opts["n"] then order := 4
if \opts["u"] then unique := 1
ccount := table(0) # table of characters
while ccount[reads()] += 1
a := sort(ccount,order)
if unique = 1 then
  while s := get(a) do if get(a) = 1 then write(s)
else {
  rwidth := 0; every rwidth <:= *!a
  while s := get(a) do
    write(left(image(s),10),right(get(a),rwidth))
  }
end

```