CSc 372
Comparative Programming Languages

36: Icon — Examples

Christian Collberg

collberg@cs.arizona.edu

Department of Computer Science
University of Arizona

Copyright © 2004 Christian Collberg
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:
Example 1 (a): Soundex

procedure soundex(name)
    local first, c, i
    name := map(name, string(&lcase), string(&ucase))
    name := map(name, "ABCDEFGHIJKLMNOPQRSTUVWXYZ", "123.12..22455.12623.1.2.2")

    # Retain the first letter of the name
    first := name[1]

    name := map(name, "ABCDEFGHIJKLMNOPQRSTUVWXYZ", "123.12..22455.12623.1.2.2")

    # Convert to uppercase.
    name := map(name, "ABCDEFGHIJKLMNOPQRSTUVWXYZ", "123.12..22455.12623.1.2.2")
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
Example 1: Soundex...

\[
\text{left}(s_1, i, s_2) \quad \text{shift } s_1 \text{ to the left, append } s_2: s \text{ until position } i \text{ is reached.}
\]

Example

\[
\begin{align*}
\text{COLLBERG} \Rightarrow \text{(code)} & \quad "2.441.62" \Rightarrow \text{(remove duplicates)} \\
& \quad "2.41.62" \Rightarrow \text{(restore first)} \quad "C.41.62" \Rightarrow \text{(delete ".")} \quad "C4162"
\end{align*}
\]

\[
\begin{align*}
\text{COLBERG} \Rightarrow \text{(code)} & \quad "2.41.62" \Rightarrow \text{(remove duplicates)} \quad "2.41.62" \\
& \Rightarrow \text{(restore first)} \quad "C.41.62" \Rightarrow \text{(delete ".")} \quad "C4162"
\end{align*}
\]
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
Example 3: Pack

# This program 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)
        while write(read(in))
    }

end
Example 4: Tablc

# 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
Example 4 (b): Tablc...  

```plaintext
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))
    }
```