

[\[Browse source program\]](#) [\[Execute source program\]](#)

```
-----  
-- Example for CGI programming in Curry: a questionnaire  
-----
```

```
import HTML
import System
import Directory
import Read
import IOExts(exclusiveIO)

-- The parameters of the form:
question = "Who is your favorite actress?"

choices = ["Doris Day","Jodie Foster","Marilyn Monroe",
           "Julia Roberts","Sharon Stone","Meryl Streep"]

-- Form for voting:
questForm :: IO HtmlForm
questForm = return $ form "Vote Form"
  ([h1 [htxt question],
    radio_main vref "0", htxt (' ':head choices), breakline] ++
   concatMap (\(i,s)->[radio_other vref (show i), htxt (' ':s), breakline])
              (zip [1..] (tail choices)) ++
   [hrule, button "submit" questHandler])
  where
    vref free

    questHandler env = do
      exclusiveIO (voteFile++".lock")
      (incNumberInFile (readNat (env vref)))
      evalForm

-- Form for evaluating current votes:
evalForm :: IO HtmlForm
evalForm = do
  votes <- exclusiveIO (voteFile++".lock") readVoteFile
  return $ form "Evaluation"
    [h1 [htxt "Current votes:"],
```

```

        table (map (\(s,v)->[[htxt s],[htxt $ show v]])
              (zip choices votes))]

-- increment n-th element in a list:
incNth :: [Int] -> Int -> [Int]
incNth [] _ = []
incNth (x:xs) n = if n==0 then (x+1):xs else x:incNth xs (n-1)

-- increment a number in the vote file:
incNumberInFile :: Int -> IO ()
incNumberInFile n = do
    initVoteFile (length choices)
    nums <- readVoteFile
    writeVoteFile (incNth nums n)

-- initialize data file if necessary:
initVoteFile :: Int -> IO ()
initVoteFile n = do
    existnumfile <- doesFileExist voteFile
    if existnumfile then done
    else writeVoteFile (take n (repeat 0))

-- read a file with lines containing numbers and return them in a list:
readVoteFile :: IO [Int]
readVoteFile = do
    vfcont <- readFile voteFile
    return (map readNat (lines vfcont))

-- write a list of numbers into a file (each number into a line):
writeVoteFile :: [Int] -> IO ()
writeVoteFile nums = do
    writeFile (voteFile++".new") (concatMap (\n->show n++"\n") nums)
    system ("mv "++voteFile++".new "++voteFile)
    done

-- the file where the current numbers of votes are stored:
voteFile = "votes.data"

```

```
-- Install the CGI program by:  
-- makecurrycgi -m questForm questionnaire
```