```
AFFIRMHelp                            10
AFFIRMSpellingCorrect                 1
appendEscape                          8
DetermineInitialRespellings           2
insertEscapes                         9
MakeSpellingList                      3
ReduceRespellingsInteractively        4
ReduceRespellingsNonInteractively     5
SpellingSynonym                       6
SuperSpell                            7
```

```
(FILECREATED "27-Mar-80 13:07:49" <AFFIRM>HELP.;6 13330

     previous date: "24-Mar-80 14:30:30" <DTHOMPSON>HELP.;10)


(PRETTYCOMPRINT HELPCOMS)

(RPAQQ HELPCOMS [(* Spelling Correction Mechanism)
                 (FNS AFFIRMSpellingCorrect DetermineInitialRespellings MakeSpellingList
                      ReduceRespellingsInteractively ReduceRespellingsNonInteractively SpellingSynonym
                      SuperSpell appendEscape insertEscapes)
                 (RECORDS SpellingGuess)
                 (VARS (DontAskJustTake 60)
                       (NonInteractiveThresholdSize 1)
                       (InteractiveThresholdSize 3)
                       (NonInteractiveRelativeAgreementList ' (55 60 70 80 90))
                       (InteractiveRelativeAgreementList ' (40 60 70 80)))
                 (* Help System)
                 (FNS AFFIRMHelp)
                 (VARS (HelpMessageGiven NIL))
                 (* BLOCK Specifications)
                 (BLOCKS (AFFIRMSpellingCorrectBLOCK AFFIRMSpellingCorrect DetermineInitialRespellings
                                                     MakeSpellingList ReduceRespellingsInteractively
                                                     ReduceRespellingsNonInteractively SpellingSynonym
                                                     SuperSpell appendEscape insertEscapes
                                                     (ENTRIES AFFIRMSpellingCorrect SpellingSynonym
                                                              SuperSpell)
                                                     (GLOBALVARS DontAskJustTake Escape
                                                                 InteractiveRelativeAgreementList
                                                                 InteractiveThresholdSize
                                                                 NonInteractiveRelativeAgreementList
                                                                 NonInteractiveThresholdSize NOSPELLFLG)
                                                     (NOLINKFNS . T])
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

```
(* Spelling Correction Mechanism) ]

(DEFINEQ

                                                                                   1

(AFFIRMSpellingCorrect
  [LAMBDA (item itemList defaultItem dontAskUser dontTellUser)     (* D.Thompson "28-Jan-80 11:30")
    (PROG (NOSPELLFLG correctedItem guesses keyList optionList question response)

          (* * assumption: A spelling error is to be corrected. Nothing has been typed to the user so far.
          (Contrast this with the assumptions of the PossiblyMisspelled? routine))


          (if itemList
            else (RETURN defaultItem))
          (itemList←(MKLIST itemList))
          (if correctedItem←(SpellingSynonym item itemList)
              then (RETURN correctedItem))
          (guesses←(DetermineInitialRespellings item itemList dontAskUser))
          (if dontAskUser
              then (RETURN (if guesses:Respellings
                                then (ReduceRespellingsNonInteractively item guesses dontTellUser)
                              else defaultItem))
            else (if guesses:Respellings
                     then (if correctedItem←(ReduceRespellingsInteractively item guesses dontTellUser)
                              then (RETURN correctedItem)
                            else (printout NIL "No, eh?" ,,))
                   else (printout NIL .TAB0 8 "what?" , item "?" ,,))

          (* * either: -
          (1) the list of initial respellings was NIL, or -
          (2) the user rejected all the choices on the initial respelling list)


                question←(CONCAT "Please retype " item ": ")
                optionList←'(EXPLAINDELIMITER ", ")
                keyList← <'(/" (None!)" CONFIRMFLG NIL) !(for i in itemList collect <(if (LISTP i)
                                                                                        then i:1
                                                                                      else i)
                                                                                     "" 'CONFIRMFLG NIL>)

                                >
                (RETURN (if correctedItem←(AFFIRMUSER NIL '/ question keyList NIL NIL optionList)='/
                            then defaultItem
                          else (SpellingSynonym correctedItem itemList])

                                                                                   2

(DetermineInitialRespellings
  [LAMBDA (item itemList dontAskUser)                              (* D.Thompson "24-Mar-80 13:39")
    (PROG (answer done reducedList respellings thresholdSize)

          (* * assumptions: item is not in itemList)


          (* * algorithm: keep shrinking list of possible respellings until: -
          (1) list of respellings shrinks to NIL (then return the last nonNIL list); OR -
          (2) list size shrinks to less than some threshold size (then return this "maximally minimal" list) -
          the threshold sizes and relative percentage matches to be used in the shrinking process are contained in
          globals)


          (* * returns: a list (possibly NIL) of respellings, of size not necessarily less than or equal to the
          threshold size; and the relative percentage figure that produced that list)


          (respellings←itemList)
          (answer←(create SpellingGuess
                            RelativeMatch ← 0
                            Respellings ← NIL))
            (thresholdSize←(if dontAskUser
                               then NonInteractiveThresholdSize
                             else InteractiveThresholdSize)+ 1)
            (for rel in (if dontAskUser
                            then NonInteractiveRelativeAgreementList
                          else InteractiveRelativeAgreementList)
               until done do (if reducedList←(MakeSpellingList (FIXSPELL item rel respellings 'NO-MESSAGE NIL
                                                                                         NIL 'EVERYTHING T))
                                 then answer:RelativeMatch←rel
                                      answer:Respellings←reducedList
                                      (if (FNTH reducedList thresholdSize)
                                          then respellings←reducedList
                                        else done←T)
                               else done←T))
            (RETURN answer])
```

3

```
(MakeSpellingList
   [LAMBDA (list)                                                      (* D.Thompson "24-Mar-80 13:33")
     (if (NLISTP list)
         then (MKLIST list)
       else (for x in list collect (if (NLISTP x)
                                        then x
                                      elseif x::1
                                        then <(firstElement x:1) !(firstElement x::1) >
                                      else (firstElement x:1])
```

4

```
(ReduceRespellingsInteractively
   [LAMBDA (item guesses dontTellUser)                                 (* D.Thompson "24-Mar-80 14:00")

           (* * This routine attempts to reduce the possible respelling act handed to it by asking the user for help.)


     (PROG (correctedItem keyList optionList question response synonym)
           (correctedItem←guesses:Respellings)
           (if (EQLENGTH correctedItem 1)
               then correctedItem←correctedItem:1
                    (if (LISTP correctedItem)
                        then                                           (* a synonym)
                             synonym←correctedItem::1
                             correctedItem←correctedItem:1
                      else synonym←correctedItem)
                    (RETURN (if (IGEQ guesses:RelativeMatch DontAskJustTake)
                                then (if dontTellUser
                                       else (printout NIL .TAB0 0 "(" item , "=>" , correctedItem ")" T))
                                     synonym
                              else

           (* * obtain assurance from the user that this singleton respelling is indeed the right one)


                                   question←(CONCAT "Do you mean " correctedItem " for " item "? ")
                                   keyList← <<'Yes (CONCAT " (" correctedItem ")") > <'No NIL 'RETURN '(QUOTE
                                                                                                         /)
                                                                                                       > <'/" (No)" >>
                                   optionList←'(AUTOCOMPLETEFLG T)
                                   (if (AFFIRMUSER NIL 'Yes question keyList NIL NIL optionList)='/
                                       then NIL
                                     else synonym)))
             else (AFFIRMMAPRINT (CONCAT "what? Please correct " item " using the list:")
                                 (for i in correctedItem collect (if (LISTP i)
                                                                     then i:1
                                                                   else i))
                                 "or" T)
                  question←(CONCAT "Please correct " item ": ")
                  optionList←'(EXPLAINDELIMITER ", ")
                  keyList← <'(/" (None!)") !(for i in correctedItem collect <(if (LISTP i)
                                                                                then i:1
                                                                              else i)
                                                                             NIL>)
                             >
                  (RETURN (if response←(AFFIRMUSER NIL '/ question keyList NIL NIL optionList)='/
                              then NIL
                            else (SpellingSynonym response correctedItem])
```

5

```
(ReduceRespellingsNonInteractively
  [LAMBDA (item guesses dontTellUser)                                 (* D.Thompson "24-Mar-80 13:57")
     (PROG (correctedItem synonym)
           (correctedItem←guesses:Respellings)
           (if (EQLENGTH correctedItem 1)
               then correctedItem←correctedItem:1
                    (if (LISTP correctedItem)
                        then synonym←correctedItem::1
                             correctedItem←correctedItem:1
                      else synonym←correctedItem)
                    (if dontTellUser
                      else (printout NIL .TAB0 0 "(" item , "=>" , correctedItem ")" T))
                    (RETURN synonym)
            else (RETURN NIL])
```

6

```
(SpellingSynonym
  [LAMBDA (item itemList extendName)                                  (* D.Thompson "17-Mar-80 20:03")

          (* * This routine ferrets out exact matches and synonyms so the spelling corrector needn't be bothered.)


     (PROG (APPROVEFLG correction)
           (RETURN (if correction←(for i in itemList thereis item=(if (NLISTP i)
                                                                        then i
                                                                      else i:1))
                      then (if (NLISTP correction)
                               then correction
                             else correction::1)
                    elseif extendName
                      then (FIXSPELL (AppendChar item Escape NIL)
                                     100 itemList 'NO-MESSAGE])
```

7

```
(SuperSpell
  [LAMBDA (misspelledWord spellingList defaultItem dontAskUser dontTellUser prefix suffix)
                                                                      (* D.Thompson " 4-Oct-79 08:57")
     (PROG (oldWord newWord respelling done)
           (prefix←(if prefix
                    else ""))
           (suffix←(if suffix
                    else ""))
           (until done for f from 1 to 4 do [newWord←(SELECTQ f
                                                                (1 misspelledWord)
                                                                (2 (appendEscape misspelledWord))
                                                                (3 (insertEscapes misspelledWord))
                                                                (appendEscape (insertEscapes misspelledWord]
                                            (if newWord~=oldWord and respelling←(AFFIRMSpellingCorrect
                                                                        (PACK* prefix newWord suffix)
                                                                        spellingList NIL T dontTellUser)
                                                then done←T
                                              else oldWord←newWord))
           (RETURN (if done
                       then respelling
                     elseif dontAskUser
                    then defaultItem
                  else (AFFIRMSpellingCorrect (PACK* prefix misspelledWord suffix)
                                                spellingList defaultItem dontAskUser dontTellUser])

                                                                                          8

(appendEscape
  [LAMBDA (word)                                            (* D.Thompson "13-Jul-79 15:58")
    (if (NTHCHAR word -1)=Escape
        then word
      else (PACK* word Escape])

                                                                                          9

(insertEscapes
  [LAMBDA (word)                                            (* D.Thompson "13-Jul-79 16:31")
    (PROG (chars)
          (chars←(DREMOVE Escape (UNPACK word)))
          (RETURN (PACK < !(if (U-CASEP chars:1)
                                 then <chars:1>
                               else <Escape chars:1>)
                           !(for c in chars::1 join (if (U-CASEP c)
                                                            then <Escape c>
                                                          else <c>))
                           >])
)
[DECLARE: EVAL@COMPILE

(RECORD SpellingGuess (RelativeMatch Respellings))
]

(RPAQ DontAskJustTake 60)

(RPAQ NonInteractiveThresholdSize 1)

(RPAQ InteractiveThresholdSize 3)

(RPAQ NonInteractiveRelativeAgreementList ' (55 60 70 80 90))

(RPAQ InteractiveRelativeAgreementList ' (40 60 70 80))
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

```
(* Help System) ]

(DEFINEQ


(AFFIRMHelp
  [LAMBDA (topic aux)                                        (* D.Thompson " 5-Feb-80 12:26")

          (* * This routine oversees the display of help information from the online database.)


    (SELECTQ (U-CASE topic)
             (EXEC (if HelpMessageGiven
                       else (printout NIL .TAB0 0 "The on-line help system is not yet functional." T
                                      "See the command synopses in the reference manual."
                                      T T)
                            HelpMessageGiven←T)
                   (print '(known Commands)))
             (PROFILE (printout NIL .TAB0 0 "sorry -- no info available!" T))
             (PROGN (printout NIL .TAB0 0 "sorry -- no info available!" T)
                    (Unexpected 'HelpTopic])
)

(RPAQ HelpMessageGiven NIL)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

(* BLOCK Specifications)]

```
(DECLARE: DONTEVALeLOAD DOEVALeCOMPILE DONTCOPY
(BLOCK: AFFIRMSpellingCorrectBLOCK AFFIRMSpellingCorrect DetermineInitialRespellings MakeSpellingList
        ReduceRespellingsInteractively ReduceRespellingsNonInteractively SpellingSynonym SuperSpell
        appendEscape insertEscapes (ENTRIES AFFIRMSpellingCorrect SpellingSynonym SuperSpell)
        (GLOBALVARS DontAskJustTake Escape InteractiveRelativeAgreementList InteractiveThresholdSize
                    NonInteractiveRelativeAgreementList NonInteractiveThresholdSize NOSPELLFLG)
        (NOLINKFNS . T))
]
(DECLARE: DONTCOPY
  (FILEMAP (NIL (1496 11479 (AFFIRMSpellingCorrect 1508 . 3511) (DetermineInitialRespellings 3515 . 5316) (
MakeSpellingList 5320 . 5781) (ReduceRespellingsInteractively 5785 . 8032) (ReduceRespellingsNonInteractively
8036 . 8729) (SpellingSynonym 8733 . 9476) (SuperSpell 9480 . 10717) (appendEscape 10721 . 10959) (
insertEscapes 10963 . 11476)) (11892 12660 (AFFIRMHelp 11904 . 12657)))))
STOP
```

```
(FILECREATED "11-Aug-80 21:34:42" <AFFIRM>INDUCTION..14 3923

     changes to:  INDUCTION Prop\Induction IH\Induction Prop\Induction\Interface IH\Induction\Interface)


(PRETTYCOMPRINT INDUCTIONCOMS)

(RPAQQ INDUCTIONCOMS [(P (CheckLoad (QUOTE TYPE)
                                    (QUOTE (118 . <AFFIRM>BASE-AFFIRM.EXE.54))
                                    (QUOTE Induction)))
                      (FNS * InductionFNS)
                      (FNS * Induction\InterfaceFNS)
                      (VARS * InductionConstants)
                      (VARS * Induction\InterfaceConstants)
                      (IFPROP ALL * InductionConstants)
                      (IFPROP (PrimaryLHSides EqualOp EQOP)
                              * InductionFNS)
                      (IFPROP (PrimaryLHSides EqualOp EQOP)
                              * Induction\InterfaceFNS)
                      (P (InitializeLoad TYPE Induction 118 ((NoteInterfaces Induction\InterfaceFNS)
                                                             (initInfix (QUOTE Induction))
                                                             (initNeeds (QUOTE Induction))
                                                             (NoteDeclarations (QUOTE Induction))
                                                             (NoteLeftHandSides InductionFNS])
(CheckLoad (QUOTE TYPE)
           (QUOTE (118 . <AFFIRM>BASE-AFFIRM.EXE.54))
           (QUOTE Induction))

(RPAQQ InductionFNS (Prop\Induction IH\Induction Equal\Induction))
(DEFINEQ

(Prop\Induction
  [LAMBDA (val)                                                       (* R.Erickson "11-Jun-80 21:11")
    (if ExpandInductors
        then (ComputeInductionExpression CurrentPropn IndVar val)
      else <PROPOP val>])

(IH\Induction
  [LAMBDA (val)                                                       (* R.Erickson "17-Jun-80 18:01")

          (* * see IH\Builtin This form occurs only in schemas.)


    (if ExpandInductors
        then <IH2OP val IndTarget>
      else <IH1OP val>])

(Equal\Induction
  [LAMBDA (x y)
    (if <'Equal\Induction x y>])
)

(RPAQQ Induction\InterfaceFNS (Prop\Induction\Interface IH\Induction\Interface))
(DEFINEQ

(Prop\Induction\Interface
  [LAMBDA (a TooManyArguments)
    (if a:1='ExpressionWithType and TooManyArguments=NIL and (Report Prop\Induction\Interface 1 interface)
        then (ExpressionWithType <'Prop\Induction a:2> Boolean)
      elseif NIL])

(IH\Induction\Interface
  [LAMBDA (a TooManyArguments)
    (if a:1='ExpressionWithType and TooManyArguments=NIL and (Report IH\Induction\Interface 1 interface)
        then (ExpressionWithType <'IH\Induction a:2> Boolean)
      elseif NIL])
)

(RPAQQ InductionConstants (Induction))

(RPAQQ Induction Induction)

(RPAQQ Induction\InterfaceConstants NIL)
  (PUTPROPS Induction IsConstant T
                    DeclaredType Induction
                    LocalDeclarations ((a\Interface ExpressionWithType a\Induction any))
                    Infix NIL
                    Needs NIL
                    EqualOp Equal\Induction
                    NoteInterface (Induction\Integer\Interface))

(RPAQQ InductionFNS (Prop\Induction IH\Induction Equal\Induction))

(PUTPROPS Prop\Induction EqualOp EQV\Boolean)

(PUTPROPS IH\Induction EqualOp EQV\Boolean)

(PUTPROPS Equal\Induction EqualOp EQV\Boolean)

(PUTPROPS Equal\Induction EQOP T)

(RPAQQ Induction\InterfaceFNS (Prop\Induction\Interface IH\Induction\Interface))

(PUTPROPS Prop\Induction\Interface PrimaryLHSides (1 (1 Prop\Induction\Interface (ExpressionWithType
                                                                                   a\Induction any)
                                                         NIL)))

(PUTPROPS IH\Induction\Interface PrimaryLHSides (1 (1 IH\Induction\Interface (ExpressionWithType a\Induction
                                                                                             any)
                                                       NIL)))
(InitializeLoad TYPE Induction 110 ((NoteInterfaces Induction\InterfaceFNS)
               (InitInfix (QUOTE Induction))
               (InitNeeds (QUOTE Induction))
               (NoteDeclarations (QUOTE Induction))
               (NoteLeftHandSides InductionFNS)))
(DECLARE: DONTCOPY
  (FILEMAP (NIL (1131 1805 (Prop\Induction 1143 . 1410) (IH\Induction 1414 . 1710) (Equal\Induction 1723 . 1802)
) (1897 2492 (Prop\Induction\Interface 1909 . 2200) (IH\Induction\Interface 2204 . 2489)))))
STOP
```