MODULE *DisasterDetection*

EXTENDS *Integers*, *Sequences*, *TLC*, *FiniteSets*

CONSTANTS *InputIndicators*, *DisasterNames*,
IndicatorsByDisaster

PlusCal algorithm for disaster identification

```
--algorithm DisasterDetection
variables DisasterMatches = [d ∈ 1 .. Len(DisasterNames) ↦ 0],
          ProbableDisaster = "", HighestMatchCount = 0, i = 1,
          disasterIndicators, newMatchCount ;
begin
 CountMatches:
  while i ≤ Len(DisasterNames) do
     Local variable assignments
    disasterIndicators := IndicatorsByDisaster[i] ;
    newMatchCount := Cardinality(disasterIndicators
                                 ∩ InputIndicators) ;
    DisasterMatches[i] := newMatchCount ;
    i := i + 1 ;
  end while ;

     Reset the counter for the next loop
    i := 1 ;

    DetermineDisaster:
        while i ≤ Len(DisasterNames) do
            if DisasterMatches[i]
                > HighestMatchCount then
                ProbableDisaster := DisasterNames[i] ;
                HighestMatchCount := DisasterMatches[i] ;
            end if ;
            i := i + 1 ;
        end while ;
print (⟨"Most Probable disaster based on indicators is: ",
        ProbableDisaster, " with ", HighestMatchCount,
        " matching indicators."⟩) ;
end algorithm
```

BEGIN TRANSLATION ($chksum(pcal) =$ "$d50f0a22$" $\wedge\ chksum(tla) =$ "$45f40e27$")

CONSTANT *defaultInitValue*

VARIABLES *DisasterMatches*, *ProbableDisaster*, *HighestMatchCount*, *i*,
disasterIndicators, *newMatchCount*, *pc*

$vars \triangleq \langle DisasterMatches, ProbableDisaster, HighestMatchCount, i,$
$disasterIndicators, newMatchCount, pc\rangle$

$Init \triangleq$ Global variables

$\land\ DisasterMatches = [d \in 1 \,..\, Len(DisasterNames) \mapsto 0]$
$\land\ ProbableDisaster = \text{“”}$
$\land\ HighestMatchCount = 0$
$\land\ i = 1$
$\land\ disasterIndicators = defaultInitValue$
$\land\ newMatchCount = defaultInitValue$
$\land\ pc = \text{“CountMatches”}$

$CountMatches \triangleq$
$\land\ pc = \text{“CountMatches”}$
$\land$ IF $i \leq Len(DisasterNames)$
THEN $\land\ disasterIndicators' = IndicatorsByDisaster[i]$
$\land\ newMatchCount' = Cardinality(disasterIndicators' \cap InputIndicators)$
$\land\ DisasterMatches' = [DisasterMatches$ EXCEPT $![i] = newMatchCount']$
$\land\ i' = i + 1$
$\land\ pc' = \text{“CountMatches”}$
ELSE $\land\ i' = 1$
$\land\ pc' = \text{“DetermineDisaster”}$
$\land$ UNCHANGED $\langle DisasterMatches, disasterIndicators, newMatchCount\rangle$
$\land$ UNCHANGED $\langle ProbableDisaster, HighestMatchCount\rangle$

$DetermineDisaster \triangleq$
$\land\ pc = \text{“DetermineDisaster”}$
$\land$ IF $i \leq Len(DisasterNames)$
THEN $\land$ IF $DisasterMatches[i] > HighestMatchCount$
THEN $\land\ ProbableDisaster' = DisasterNames[i]$
$\land\ HighestMatchCount' = DisasterMatches[i]$
ELSE $\land$ TRUE
$\land$ UNCHANGED $\langle ProbableDisaster, HighestMatchCount\rangle$
$\land\ i' = i + 1$
$\land\ pc' = \text{“DetermineDisaster”}$
ELSE $\land\ PrintT((\langle \text{“Most Probable disaster based on indicators is: ”}, ProbableDisaster, \text{“ with ”}, HighestMatchCount, \text{“ matching indicators.”}\rangle))$
$\land\ pc' = \text{“Done”}$
$\land$ UNCHANGED $\langle ProbableDisaster, HighestMatchCount, i\rangle$
$\land$ UNCHANGED $\langle DisasterMatches, disasterIndicators, newMatchCount\rangle$

Allow infinite stuttering to prevent deadlock on termination.
$Terminating \triangleq\ pc = \text{“Done”} \land$ UNCHANGED $vars$

$Next \triangleq CountMatches \lor DetermineDisaster \lor Terminating$

$Spec \triangleq Init \land \Box[Next]_{vars}$

$Termination \triangleq \Diamond(pc = \text{“Done”})$

END TRANSLATION