Task 12.05

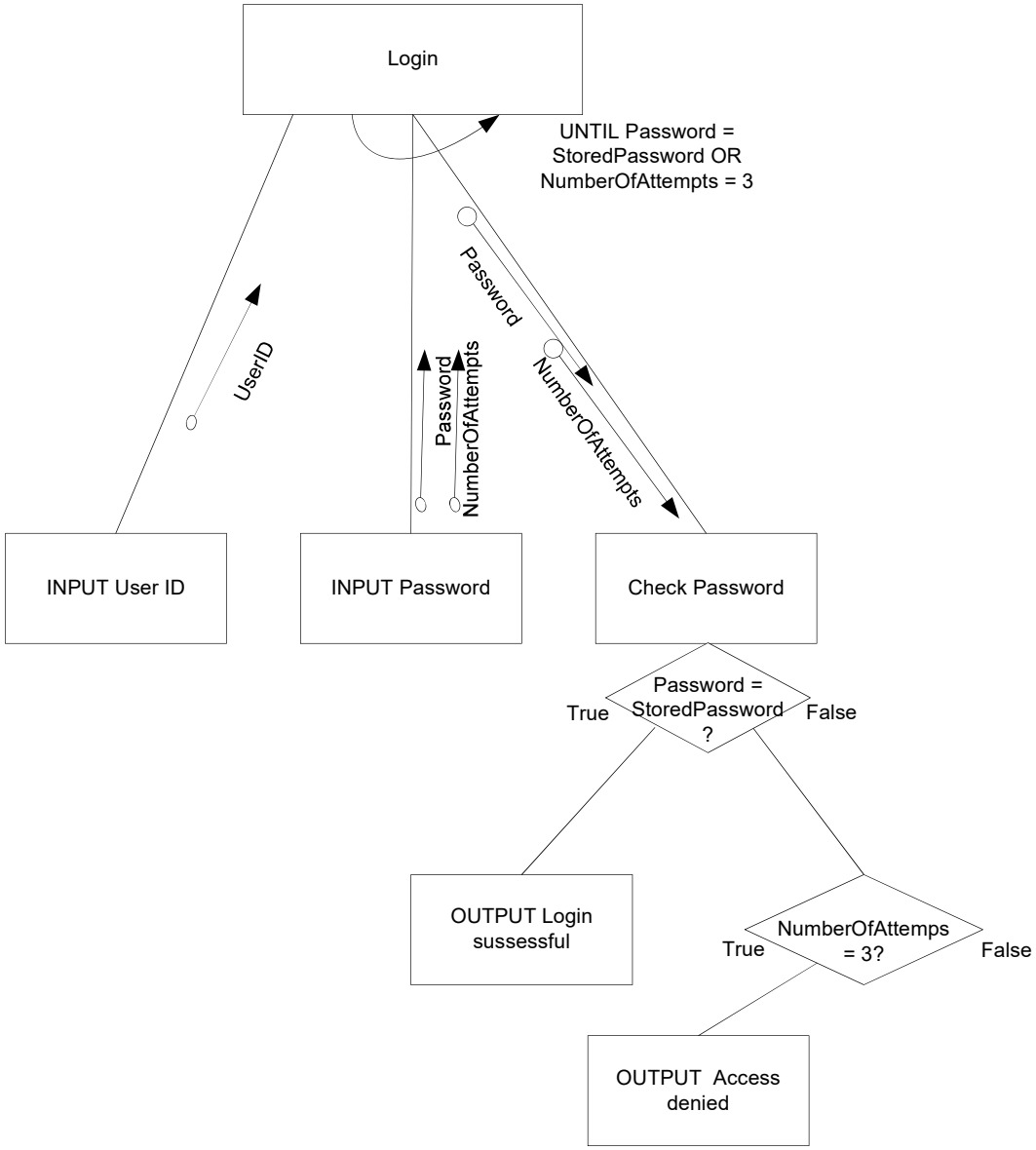


Figure 12.03

Exam-style questions in Chapter 12

1 a A: Initialise Tally

B: Generate random number

C: RandomNumber

D: Tally

E: Tally

b Pseudocode for random number tally

```
DECLARE Tally : ARRAY[1..20] OF INTEGER
CALL InitialiseTally(Tally)
FOR Count ← 1 TO NumberOfTests
```

```
   RandomNumber ← GenerateRandomNumber(20)
   CALL UpdateTally(RandomNumber, Tally)
ENDFOR
CALL OutputTally(Tally)
```

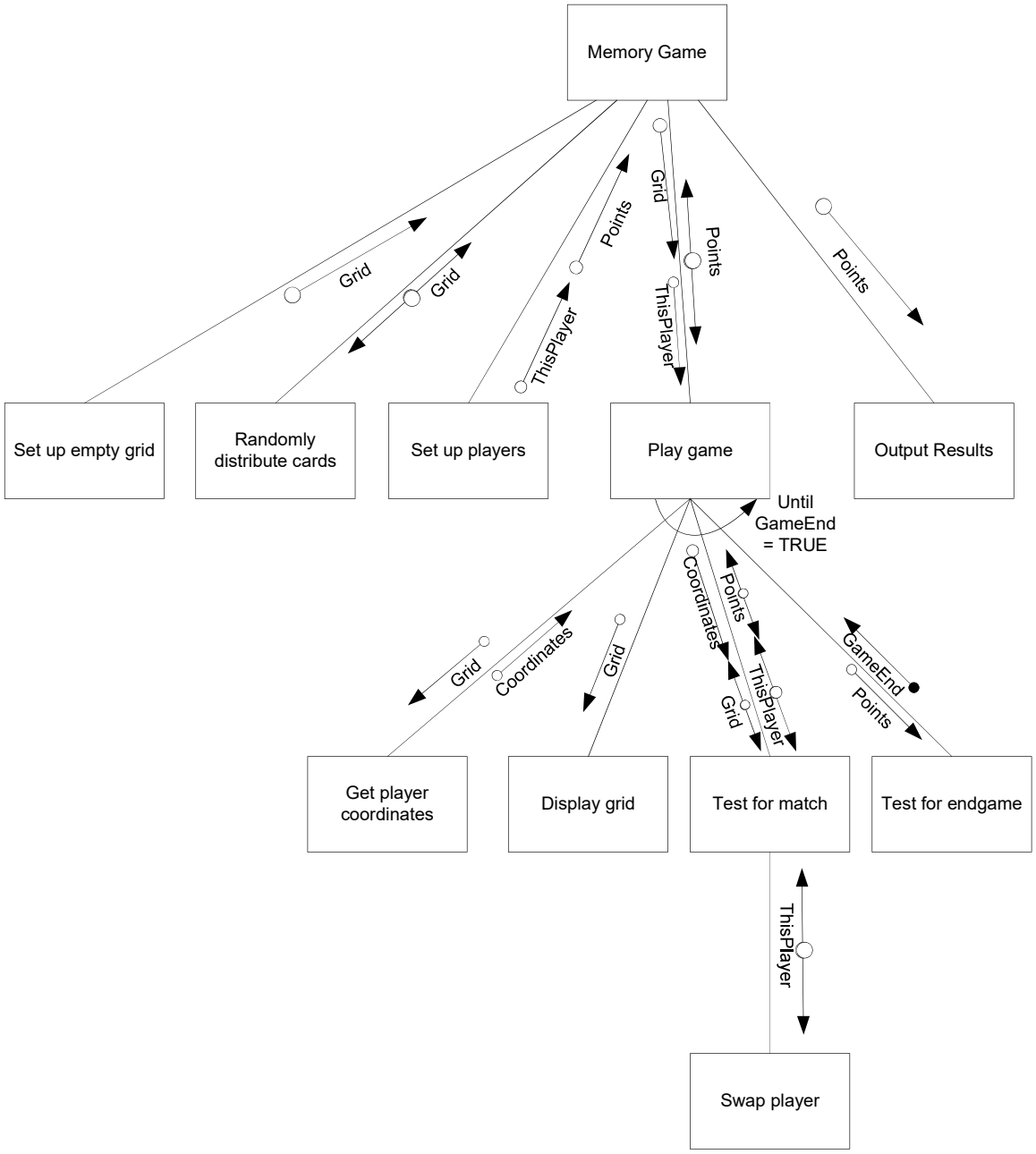
2 a Step-wise refinement

b

```
PROCEDURE SetUpEmptyGrid
   FOR i ← 1 TO 8
      FOR j ← 1 TO 8
         Grid[i, j] ← 0
      ENDFOR
   ENDFOR
ENDPROCEDURE

PROCEDURE RandomlyDistributeCards
   FOR Number ← 1 TO 32
      CALL GetEmptyGridPosition
      Grid[x, y] ← Number
      CALL GetEmptyGridPosition
      Grid[x, y] ← Number
   ENDFOR
ENDPROCEDURE

PROCEDURE GetEmptyGridPosition
   REPEAT
      x ← RandomNumber(1,8)
      y ← RandomNumber(1,8)
   UNTIL Grid[x, y] = 0 // find a grid position without a card
ENDPROCEDURE

PROCEDURE SetUpPlayers
   Points[1] ← 0
   Points[2] ← 0
   ThisPlayer ← 1
ENDPROCEDURE

PROCEDURE GetPlayerCoordinates
   REPEAT
      INPUT x1, y1
   UNTIL Grid[x1, y1] > 0 // check grid position has a card
   CALL DisplayGrid
   REPEAT
      INPUT x2, y2
           // check grid position has a card and is not the
same as first card
   UNTIL (Grid[x2, y2] > 0) AND ((x1 <> x2) OR (y1 <> y2))
ENDPROCEDURE

PROCEDURE DisplayGrid
   FOR i ← 1 TO 8
      FOR j ← 1 TO 8
         IF (I = x1) AND (j = y1) // it is the chosen card
            THEN
               OUTPUT Grid[i, j]
```

```
            ELSE
               IF Grid[I, j] = 0 // the card in this position
has been removed
                  THEN
                     OUTPUT '   '
                  ELSE   // back of card to be shown as ' ? '
                     OUTPUT ' ? '
               ENDIF
         ENDIF
      ENDFOR
   ENDFOR
ENDPROCEDURE

PROCEDURE TestForMatch
   IF Grid[x1, y1] = Grid[x2, y2]
      THEN
         // match found, remove cards
         Grid[x1, y1] ← 0
         Grid[x2, y2] ← 0
         // increment points
         Points[ThisPlayer] ← Points[ThisPlayer] + 1
      ELSE
         CALL SwapPlayers
   ENDIF
ENDPROCEDURE

PROCEDURE SwapPlayers
   IF ThisPlayer = 1
      THEN
         ThisPlayer ← 2
      ELSE
         ThisPlayer ← 1
   ENDIF
ENDPROCEDURE

PROCEDURE TestForEndGame
   IF Points[1] + Points[2] = 32
      THEN
         GameEnd ← TRUE
   ENDIF
ENDPROCEDURE

PROCEDURE OutputResults
   OUTPUT Points[1]
   OUTPUT Points[2]
ENDPROCEDURE
```

C



D

Figure 12.04

Task 14.01

```
PROCEDURE SetValues
   INPUT Symbol
   CALL InputMaxNumberOfSymbols // need to ensure it is an odd
number
   NumberOfSpaces ← (MaxNumberOfSymbols -1) / 2
   NumberOfSymbols ← 1
ENDPROCEDURE

PROCEDURE InputMaxNumberOfSymbols
```