

```

// Global Arrays
DECLARE studentNames: ARRAY[1:15] OF STRING
DECLARE ArrMaths: ARRAY[1:15] OF REAL
DECLARE ArrPhysics: ARRAY[1:15] OF REAL

// Library routines declearation
FUNCTION pow(num:REAL, p:INTEGER) RETURNS REAL
    IF p = 0 THEN
        RETURN 1
    ENDIF
    RETURN num*pow(num, p-1)
ENDFUNCTION
FUNCTION ROUND(num:REAL, dp:INTEGER) RETURNS REAL
    RETURN ((num*pow(10, dp)) DIV 1) / pow(10, dp)
ENDFUNCTION
FUNCTION MYRANDOM() RETURNS REAL
    RETURN RAND(1)
ENDFUNCTION

// Procedure to populate arrays
PROCEDURE PopulateArrays()
    DECLARE i:INTEGER
    FOR i <- 1 TO 15
        OUTPUT "Enter name of student ", i
        INPUT studentNames[i]
        ArrMaths[i] <- MYRANDOM()*100
        ArrPhysics[i] <- MYRANDOM()*100
        // ArrMaths[i] <- i
        // ArrPhysics[i] <- i+1
    NEXT i
ENDPROCEDURE

// Function to calculate average for an array
// FUNCTION CalculateAverage(subjectMarks: ARRAY[1:15] OF INTEGER) RETURNS
REAL
//     DECLARE total: REAL
//     DECLARE average: REAL

```

```

//      total <- 0
//      DECLARE i:INTEGER
//      FOR i <- 1 TO 15
//          total <- total + subjectMarks[i]
//      NEXT i
//      average <- total / 15
//      RETURN average
// ENDFUNCTION

// Function to calculate average for math array
FUNCTION CalculateMathAverage() RETURNS REAL
    DECLARE total: REAL
    DECLARE average: REAL
    total <- 0
    DECLARE i:INTEGER
    FOR i <- 1 TO 15
        total <- total + ArrMaths[i]
    NEXT i
    average <- total / 15
    RETURN average
ENDFUNCTION

// Function to calculate average for physic array
FUNCTION CalculatePhysAverage() RETURNS REAL
    DECLARE total: REAL
    DECLARE average: REAL
    total <- 0
    DECLARE i:INTEGER
    FOR i <- 1 TO 15
        total <- total + ArrPhysics[i]
    NEXT i
    average <- total / 15
    RETURN average
ENDFUNCTION

// Procedure to display student names with average marks
PROCEDURE DisplayAverages()
    DECLARE i:INTEGER
    DECLARE averageMarks: REAL
    FOR i <- 1 TO 15
        averageMarks <- (ArrMaths[i] + ArrPhysics[i]) / 2
    
```

```
        OUTPUT "Student: ", studentNames[i], ", Average Marks: ",  
ROUND(averageMarks, 2)  
    NEXT i  
ENDPROCEDURE  
  
CALL PopulateArrays()  
CALL DisplayAverages()  
  
OUTPUT "Average mark for Mathematics: ", ROUND(CalculateMathAverage(), 1)  
OUTPUT "Average mark for Physics: ", ROUND(CalculatePhysAverage(), 1)
```