MATH5061: Week 11 Assignment

Instructions

Please submit answers to this assignment as a plain text file via email to **math5061@temple.edu**. Make sure to use the subject line (without quotes)
"MATH5061:Assignment 06:ACCESSID"
Where ACCESSID is your AccessNet ID, for example tue86537

Programming in Fortran

Task 1

Complete the subroutine isort in the following Fortran program by implementing an insertion sort algorithm on the array list which is of dimension n. On return from the subroutine, list should contain the sorted data.

Additional information:

- the Fortran command equivalent to the "break" command in Python or C++ is called EXIT
- the Fortran equivalent of the command "continue" is called CYCLE.
- the subroutine RANDOM_NUMBER() assigns a uniformly distributed random floating point number between 0 and 1.0.

```fortran
SUBROUTINE isort(list,n)
  IMPLICIT NONE
  INTEGER, INTENT(inout) :: list(*)
  INTEGER, INTENT(in) :: n

  ! implement an insertion sort for the array 'list' here

END SUBROUTINE isort

PROGRAM sortme
  IMPLICIT NONE
  INTEGER, PARAMETER :: m = 100
  REAL               :: rn
  INTEGER            :: i, vals(m)
  ! fill array with random integer numbers between 0 and 999
  CALL RANDOM_SEED()
  DO i=1,m
      CALL RANDOM_NUMBER(rn)
      vals(i) = INT(1000.0*rn)
  END DO

  WRITE(6,*) 'unsorted:'
  WRITE(6,fmt='(10I6)') vals

  CALL isort(vals,m)

  WRITE(6,*) 'sorted:'
  WRITE(6,fmt='(10I6)') vals

END PROGRAM sortme
```

Task 2

Complete the following Fortran program as indicated by the various comments.

```
  ! implement a function dotp(x,y,n) here which computes the
  ! dot product of the 1-dimensional floating point arrays x
  ! and y of the same length n

PROGRAM dot
  IMPLICIT NONE
  INTEGER             :: m
  REAL,ALLOCATABLE :: v1(:),v2(:)
  REAL                :: dotp

  PRINT*,'Give length of arrays (max. 100):'
  READ*, m
  IF ((m < 1) .OR. (m > 100)) THEN
      STOP 'Invalid length'
  END IF

  ALLOCATE(v1(m),v2(m))
  CALL RANDOM_SEED()

  ! add code here to fill both arrays with random
  ! floating point numbers between -10.0 and 10.0

  ! add code here to output both arrays

  ! add code here to output the dot product of both arrays

  ! reference result for comparison
  WRITE (6,*) 'dot_product = ', dot_product(v1,v2)

  DEALLOCATE(v1,v2)
END PROGRAM dot
```