

**NAME**

DDOTQ – compute the dot product of two vectors using extra-precision accumulation

**SYNOPSIS**

**XTY=DDOTQ(X,Y,N)**

X(N) is a REAL\*8 vector  
 Y(N) is a REAL\*8 vector  
 N is the INTEGER\*4 number of elements in X and Y  
 XTY is the REAL\*8 dot product of X and Y

**DESCRIPTION**

If N is nonpositive the routine returns XTY=0.D0. Otherwise it calls MPYACC to compute each product  $X(J)*Y(J)$  and accumulate their sum in a REAL\*8 two-part value. When all of the partial products have been accumulated it returns the sum of the two-part values as DDOTQ.

**SEE ALSO**

DDOT, which computes the dot product using double-precision accumulation

**LINKAGE**

gfortran source.f -L\${HOME}/lib -lmisc

**AUTHOR**

Michael Kupferschmid

**EXAMPLE**

```
REAL*8 X(101), Y(101), DDOTQ, ANSQ
X(1)=1.D+08
Y(1)=1.D+08
DO 1 J=2, 101
    X(J)=DFLOAT(J-1)
    Y(J)=1.D0/DFLOAT(J-1)
1 CONTINUE
ANSQ=DDOTQ(X, Y, 101)
WRITE(6, 901) ANSQ
901 FORMAT('DDOTQ finds ', 1PD23.16)
STOP
END
```

This example produced the following output:

```
unix[1] a.out
DDOTQ finds  1.00000000000000100D+16
unix[2]
```

**REFERENCES**

Kupferschmid, Michael, Classical Fortran: Programming for Engineering and Scientific Applications, CRC Press (2009), additional section 18.4+.