

Ponavljanje za kolokvij

Imamo niz od 7 elemenata gdje su elementi zadani kao $a_i = i - 1$

Odredi sumu elemenata niza od 2-gog do 5-tog člana koristeći internu fortransku funkciju SUM (ne koristi do petlju!)

```
PROGRAM niz
  INTEGER, DIMENSION(7) :: a
  a=(/(i-1,i=1,7)/)
  PRINT *,SUM(a(2:5))
END PROGRAM
```

Zadan je jednodimenzionalni niz od 7 elemenata $a = 1, 2, 1, 2, 3, 2, 5$
Korištenjem WHERE ELSEWHERE pronađi niz b za koji vrijedi
 $b = a$ za $a > 2$, inace $b = 1$

Pronađi sumu elemenata niza b korištenjem funkcije SUM

```
PROGRAM niz1
  INTEGER, DIMENSION(7) :: a,b
  DATA a / 1,2,1,2,3,2,5 /
  WHERE (a .gt. 2)
    b = a
  ELSEWHERE
    b = 1
  ENDWHERE
  PRINT *,b
  PRINT *,SUM(b)
END PROGRAM
```

Vektorski indeksi nizova

$$aa = \begin{pmatrix} 1 & 3 & 4 \\ 2 & 3 & 5 \\ 2 & 4 & 5 \end{pmatrix}$$

Vektorski indeksi $ii1 = (/3,1,2 /)$
 $ii2 = (/2,3,1/)$

Pronađi koliko je $aa=aa(ii1,ii2)$
 $bb=aa(ii2,ii1)$

Pronađi koliko je $sum(aa)$

```
program aaa
INTEGER, DIMENSION(3,3) :: aa,bb
INTEGER, DIMENSION(3) :: ii1,ii2
aa = RESHAPE((/1,2,2,3,3,4,4,5,5/),(/3,3/))
ii1 = (/3,1,2 /)
ii2 = (/2,3,1/)
aa = aa ( ii1,ii2)
bb = aa ( ii2,ii1 )
Do i=1,3
Print *,aa(i,1:3)
End do
Do i=1,3
print *, bb(i,1:3)
End do
print *,sum(aa)
end program
```

Napraviti program koji će napraviti datoteku 'unformat.dat' (na računalu 'dominis') i u nju upisati 5 realnih brojeva (od 4 byta) u binarnom, neformatiranom obliku.

Nakon toga pročitati iz datoteke zadnja tri zapisa od kraja prema naprijed. Dužina zapisa je 1.

```
PROGRAM neformatirana
CHARACTER (LEN=20) :: datoteka = 'unformat.dat'
INTEGER :: i, ierr
REAL,DIMENSION(5) :: podatak
podatak=(/1.,3.,5.,7.,9./)
OPEN(UNIT=15,FILE=TRIM(datoteka),STATUS='unknown',ACTION='write',&
ACCESS='direct',FORM='unformatted',RECL=1,ERR=20,IOSTAT=ierr)
DO i=1,5
WRITE(15,REC=i) podatak(i)
END DO
CLOSE (15)
OPEN (UNIT=15,FILE=TRIM(datoteka),STATUS='old',ACTION='read',ACCESS='direct',&
FORM='unformatted',RECL=1,ERR=20,IOSTAT=ierr)
do i=3,5,1
READ (15,REC=i) podatak(i)
PRINT *,podatak(i)
end do
CLOSE (15)
20 IF (ierr /= 0) THEN
PRINT *,'Greska kod otvaranja: ',ierr
STOP
END IF
END PROGRAM
```

Napisati modul **matrice** u kojem je definirana unarna operacija izračuna sume kvadrata dijagonalnih elemenata matrice:

$$b = \text{.tragkvadrata.a}$$

gdje je a neka matrica zadana s brojevima jednostruke točnosti, a b je suma kvadrata dijagonalnih elemenata.

$$a_{ij} = i+j \text{ za } i \text{ različito od } j, i,j=1,5$$

$$a_{ii} = i**2 \text{ za } i=1,5$$

```

module matrice
  implicit none

  interface operator ( .tragkvadrata. )
    module procedure trag
  end interface

contains
  function trag(a) result(b)
    real,intent(in), dimension(:,:) :: a
    integer :: i = 1
    real:: b
    do i=1,size(a,1)
      b = b + a(i,i)**2
    end do
  end function
end module

```

```

program kvtrag
  use matrice
  implicit none

  real, dimension(5,5):: a
  real:: b
  integer:: i,j

  do i=1,5
    do j=1,5
      if(j==i)then
        a(i,j)=i**2
      else
        a(i,j)=i+j
      endif
    end do
  end do

  b = .tragkvadrata.a
  print *, b
end program

```

Napisati modul **cetverokuti**. Unutar modula treba definirati tip podatka cetverokut kao cetvorku *tocaka*, gdje je *tocka* također novi tip podatka zadan s parom (x,y) koordinata u ravnini. Unutar modula definirati unarnu operaciju na tipu podatka cetverokut koja izračunava opseg cetverokuta: $O = .opseg.t$, gdje je t tip podatka cetverokut.

Koristeći navedenu operaciju izračunati opseg cetverokuta kojem su vrhovi u točkama (0.0,0.0), (1.0,0.0) i (1.0,1.0), (0.0,1.0)

```

Module cetverokuti
    implicit none
    type tocka
        real :: x,y
    end type
    type cetverokut
        type(tocka) :: a,b,c,d
    end type
    interface operator (.opseg.)
        module procedure proc
    end interface
contains
function proc(t) result(y)
    type(cetverokut), intent(in) :: t
    real ::a,b,c,d,y
    a = sqrt((t%a%x-t%b%x)**2 + (t%a%y-t%b%y)**2)
    b = sqrt((t%b%x-t%c%x)**2 + (t%b%y-t%c%y)**2)
    c = sqrt((t%c%x-t%d%x)**2 + (t%c%y-t%d%y)**2)
    d = sqrt((t%d%x-t%a%x)**2 + (t%d%y-t%a%y)**2)
    y= a+b+c+d
end function
end module

```

```

Program tr
    use cetverokuti
    type(cetverokut) :: t
    t%a%x=0.0
    t%a%y=0.0
    t%b%x=1.0
    t%b%y=0.0
    t%c%x=1.0
    t%c%y=1.0
    t%d%x=0.0
    t%d%y=1.0
    print *,.opseg.t
end program

```

Napisati modul **kva** u kojem treba definirati novi tip brojeva, tz. kvaternione koji predstavljaju poopćenje kompleksnih brojeva

$$\text{kvaternion}(a,b,c,d) = \begin{pmatrix} a + i * b & -d + i * c \\ d + i * c & a - i * b \end{pmatrix}$$

Neka su a , b , c i d realni brojevi dvostruke točnosti.

Za razliku od običnih kompleksnih brojeva, kvaternioni više nalikuju matricama jer ne vrijedi zakon komutacije kod množenja ili dijeljenja. Operacije množenja i dijeljenja kvaterniona (a,b,c,d) ekvivalentne su množenju i djeljenju matrica

Unutar modula treba definirati operaciju množenja kvaterniona s kvaternionom

Definirati također funkciju apsolutne vrijednosti kvaterniona, ABS, koja kao rezultat vraća realni broj jednak:

$$\text{ABS}(\text{kvaternion}(a,b,c,d)) = \text{SQRT}(a^{**2} + b^{**2} + c^{**2} + d^{**2})$$

Module kva $\begin{pmatrix} a+i*b & -d+i*c \\ d+i*c & a-i*b \end{pmatrix}$
 Implicit none

type kvaternion
 real(kind=8):: a,b,c,d
 end type

interface operator (*)
 module procedure mn_kvaternion
 end interface

interface operator (.abs.)
 module procedure absol
 end interface

contains
 Function mn_kvaternion(x,y) result(z)
 Type(kvaternion), intent(in):: x, y
 Type(kvaternion):: z
 Complex(kind=8), dimension(2,2):: f,g,h
 f(1,1:2)=(/cmplx(x%a,x%b),cmplx(-x%d,x%c)/)
 f(2,1:2)=(/cmplx(x%d,x%c),cmplx(x%a,-x%b)/)
 g(1,1:2)=(/cmplx(y%a,y%b),cmplx(-y%d,y%c)/)
 g(2,1:2)=(/cmplx(y%d,y%c),cmplx(y%a,-y%b)/)
 h=matmul(f,g)

z%a=real(h(1,1))
 z%b=aimag(h(1,1))
 z%c=aimag(h(2,1))
 z%d=real(h(2,1))
 End function
 Function absol(x) result(y)
 type(kvaternion), intent(in):: x
 real(kind=8):: y
 y=sqrt(x%a**2+x%b**2+x%c**2+x%d**2)
 end function
 end module