

---

**Εισαγωγή στους Υπολογιστές και στον Προγραμματισμό με FORTRAN**


---

**Θέμα 1<sup>ο</sup>:** (1 μονάδα)

Ελέγξτε αν το παρακάτω πρόγραμμα είναι σωστό. Αν είναι σωστό, δικαιολογείστε τα αποτελέσματα που θα προκύψουν. Αλλιώς, διορθώστε το και δικαιολογείστε τα αποτελέσματα που θα προκύψουν.

```
program themal
implicit none
integer (kind=4) :: n,i,k
n=7
do i=1,n
  k=i/2
  l=2k+i
  print * i, k, l
end do
end program themal
```

**Λύση:** Ο κώδικας έχει τα εξής σφάλματα:

- α) Δεν δηλώνεται ο τύπος της μεταβλητής  $l$  (υπάρχει προηγουμένως `implicit none`).
- β) Στην εντολή `l=2k+i` θα έπρεπε να υπάρχει κάποιος τελεστής ανάμεσα στα αντικείμενα  $2$  και  $k$ .
- γ) Στην εντολή `print * i, k, l` λείπει ένα `,` μετά το `*`.

Αν υποθέσουμε ότι ο προγραμματιστής γράφοντας `2k` εννοούσε το `2*k`, ο κώδικας διορθώνεται ως εξής:

```
program themal
implicit none
integer (kind=4) :: n,i,k,l
n=7
do i=1,n
  k=i/2
  l=2*k+i
  print *, i, k, l
end do
end program themal
```

Τα αποτελέσματα θα είναι οι αντίστοιχες τιμές των  $i$ ,  $k$ ,  $l$ :

```
1 0 1
2 1 3
3 1 4
4 2 8
5 2 9
6 3 12
7 3 13
```

καθώς η διαίρεση  $i/2$  είναι ακέραια.

**Θέμα 2<sup>ο</sup>:** (3 μονάδες)

Να γράψετε πρόγραμμα σε Fortran 90 το οποίο θα υπολογίζει και θα εκτυπώνει το παρακάτω άθροισμα:

$$S = \frac{x}{3} + \frac{x^3}{5} + \frac{x^5}{7} + \dots + \frac{x^{2n+1}}{2n+3}$$

για  $n = 100$  και  $x$  οποιονδήποτε εισαγόμενο πραγματικό αριθμό.

**Λύση:** Το  $S$  γράφεται ως εξής:

$$S = \frac{x}{3} + \frac{x^3}{5} + \frac{x^5}{7} + \dots + \frac{x^{2n+1}}{2n+3} = \sum_{i=0}^n \frac{x^{2i+1}}{2i+3}$$

Οπότε το πρόγραμμα θα είναι:

```

program thema2
implicit none
real :: s,x
integer:: n,i
n=100
read *,x
s=0.0
do i=0,n
    s=s+x**(2*i+1)/(2*i+3)
enddo
print *,s
end program thema2

```

**Θέμα 3<sup>ο</sup>:** (3 μονάδες)

a) Να γράψετε μια function  $f(x)$ , η οποία θα υπολογίζει την τιμή της συνάρτησης:

$$f(x) = \begin{cases} x^{5/3} \sqrt{x+1} - \sin x & \text{αν } x > 0 \\ -x + x^2 & \text{αν } x < 0 \\ 3 & \text{αν } x = 0 \end{cases}$$

b) Να γράψετε πρόγραμμα σε Fortran 90, το οποίο θα υπολογίζει και θα εκτυπώνει την τιμή της παραπάνω συνάρτησης για  $x \in [-1, 27]$  και βήμα 2.

**Λύση:**

a)

```

real function f(x)
implicit none
real:: x
if (x>0) then
    f=x**(5.0/3.0)*sqrt(x+1.0)-sin(x)
else if (x<0) then
    f=-x+x**2
else
    f=3.0
endif
end function f

```

c) Η επανάληψή μας πρέπει να κάνει  $(27-(-1))/2+1=15$  βήματα.

```

program thema3
implicit none
real:: x,f
integer:: i
x=-13.0
do i=1,15
    print *,i,x,f(x)
    x=x+2.0
enddo
end program thema3

```

Μπορούμε, βέβαια, να βάλουμε το πρόγραμμα να υπολογίζει το πλήθος των βημάτων δηλώνοντας επιπλέον ότι

```
integer:: n
```

και αντικαθιστώντας την εντολή do με τις

```
n=nint((27.0-(-1.0))/2.0)+1
do i=1,n
```

**Θέμα 4<sup>ο</sup>:** (3 μονάδες)

- a) Να γράψετε σε Fortran 90 μια υπορουτίνα που, αν της μεταβιβάσουμε δύο πίνακες  $A, B$  διαστάσεων  $n \times m$  με στοιχεία πραγματικούς αριθμούς, θα παράγει και θα επιστρέφει έναν άλλο πίνακα  $C$  διαστάσεων  $n \times m$  που το στοιχείο του  $C_{ij}$  θα είναι ίσο με το μέγιστο των στοιχείων των αντίστοιχων στοιχείων των  $A, B$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, m$ .
- b) Γράψτε μια κατάλληλη εντολή ενός κυρίου προγράμματος που να αξιοποιεί την παραπάνω υπορουτίνα για δύο πίνακες  $X, Y$  διαστάσεων  $6 \times 5$ .

**Λύση:**

a)

```
subroutine matrix_max_elements(n,m,a,b,c)
implicit none
integer,          intent(in)  :: n,m
real, dimension(n,m), intent(in) :: a,b
real, dimension(n,m), intent(out) :: c
integer          :: i,j
do i=1,n
  do j=1,m
    if (a(i,j)>b(i,j)) then
      c(i,j)=a(i,j)
    else
      c(i,j)=b(i,j)
    endif
  enddo
enddo
end subroutine matrix_max_elements
```

b)

```
call matrix_max_elements(6,5,x,y,z)
```

