

# ΤΡΙΤΗ ΔΙΑΛΕΞΗ

# Εντολή επανάληψης

γιατί είναι χρήσιμη;

Το άθροισμα των ακέραιων αριθμών 1–5 υπολογίζεται σε μια ακέραια μεταβλητή  $s$  με την εντολή

$$s=1+2+3+4+5$$

## Πρόβλημα

Πώς θα υπολογίσουμε το άθροισμα των ακέραιων 1–5000;

# Ισοδύναμοι υπολογισμοί του $s=1+2+3+4+5$

$$s = 0$$

$$s = s + 1$$

$$s = s + 2$$

$$s = s + 3$$

$$s = s + 4$$

$$s = s + 5$$

# Ισοδύναμοι υπολογισμοί του $s=1+2+3+4+5$

$s = 0$

$s = s + 1$

$s = s + 2$

$s = s + 3$

$s = s + 4$

$s = s + 5$

**INTEGER** :: i

$s = 0$

$i = 1$

$s = s + i$

$i = 2$

$s = s + i$

$i = 3$

$s = s + i$

$i = 4$

$s = s + i$

$i = 5$

$s = s + i$

# Ισοδύναμοι υπολογισμοί του $s=1+2+3+4+5$

```
s = 0
```

```
s = s + 1
```

```
s = s + 2
```

```
s = s + 3
```

```
s = s + 4
```

```
s = s + 5
```

```
INTEGER :: i
```

```
s = 0
```

```
i = 1
```

```
s = s + i
```

```
i = 2
```

```
s = s + i
```

```
i = 3
```

```
s = s + i
```

```
i = 4
```

```
s = s + i
```

```
i = 5
```

```
s = s + i
```

```
INTEGER :: i
```

```
s = 0
```

```
DO i = 1, 5
```

```
    s = s + i
```

```
END DO
```

## Παράδειγμα

Εκτύπωση των αριθμών 1-5 στην οθόνη:

```
PRINT *, 1
```

```
PRINT *, 2
```

```
PRINT *, 3
```

```
PRINT *, 4
```

```
PRINT *, 5
```

## Παράδειγμα

Εκτύπωση των αριθμών 1-5 στην οθόνη:

```
PRINT *, 1
```

```
PRINT *, 2
```

```
PRINT *, 3
```

```
PRINT *, 4
```

```
PRINT *, 5
```

```
INTEGER :: i
```

```
i = 1
```

```
PRINT *, i
```

```
i = 2
```

```
PRINT *, i
```

```
i = 3
```

```
PRINT *, i
```

```
i = 4
```

```
PRINT *, i
```

```
i = 5
```

```
PRINT *, i
```

## Παράδειγμα

Εκτύπωση των αριθμών 1-5 στην οθόνη:

```
PRINT *, 1
```

```
PRINT *, 2
```

```
PRINT *, 3
```

```
PRINT *, 4
```

```
PRINT *, 5
```

```
INTEGER :: i
```

```
i = 1
```

```
PRINT *, i
```

```
i = 2
```

```
PRINT *, i
```

```
i = 3
```

```
PRINT *, i
```

```
i = 4
```

```
PRINT *, i
```

```
i = 5
```

```
PRINT *, i
```

```
INTEGER :: i
```

```
DO i = 1, 5
```

```
    PRINT *, i
```

```
END DO
```



# Εντολή επανάληψης DO για συγκεκριμένο αριθμό επαναλήψεων

Σύνταξη

*DO* μεταβλητή = αρχική τιμή, τελική τιμή, βήμα αύξησης  
εντολές

*END DO*

όπου

- Η «μεταβλητή» είναι **ακέραια**.
- Τα «αρχική τιμή», «τελική τιμή», «βήμα αύξησης» είναι **ακέραιες** ποσότητες (σταθερές ή μεταβλητές ή εκφράσεις).
- Το «βήμα αύξησης» είναι θετικό ή αρνητικό. Μπορεί να παραλείπεται (και θεωρείται 1).
- Η «μεταβλητή» δεν επιτρέπεται να αλλάξει τιμή μέσα στο σώμα.

# Εντολή επανάληψης DO

## για συγκεκριμένο αριθμό επαναλήψεων

Εκτέλεση

*DO μεταβλητή = αρχική τιμή, τελική τιμή, βήμα αύξησης*  
*εντολές*

**END DO**

1. η «μεταβλητή» αποκτά την «αρχική τιμή».
2. ελέγχεται αν με διαδοχικές προσθέσεις του «βήματος αύξησης» στη «μεταβλητή» μπορούμε να φτάσουμε ή να ξεπεράσουμε την «τελική τιμή».
  - Αν όχι, διακόπτεται η επανάληψη.
  - Αν ναι, εκτελούνται οι εντολές στο σώμα.
3. Προστίθεται στη «μεταβλητή» το «βήμα αύξησης» και επαναλαμβάνεται το βήμα 2.

## Παραδείγματα επανάληψης (1/3)

Το γινόμενο των άρτιων ακέραιων αριθμών μεταξύ 2–8

( $p = 2 * 4 * 6 * 8$ ):

## Παραδείγματα επανάληψης (1/3)

Το γινόμενο των άρτιων ακέραιων αριθμών μεταξύ 2–8  
( $p = 2 * 4 * 6 * 8$ ):

```
INTEGER :: i
```

```
p = 1
```

```
i = 2
```

```
p = p * i
```

```
i = i + 2
```

```
p = p * i
```

```
i = i + 2
```

```
p = p * i
```

```
i = i + 2
```

```
p = p * i
```

## Παραδείγματα επανάληψης (1/3)

Το γινόμενο των άρτιων ακέραιων αριθμών μεταξύ 2–8  
( $p = 2 * 4 * 6 * 8$ ):

```
INTEGER :: i
p = 1
i = 2
p = p * i
i = i + 2
p = p * i
i = i + 2
p = p * i
i = i + 2
p = p * i
```

```
INTEGER :: i
p = 1
DO i = 2, 8, 2
    p = p * i
END DO
```

## Παραδείγματα επανάληψης (2/3)

- Άθροισμα των ακέραιων αριθμών από το 1 ως το 100:

```
INTEGER :: s, i
```

```
s = 0
```

```
DO i = 1, 100
```

```
    s = s + i
```

```
END DO
```

## Παραδείγματα επανάληψης (2/3)

- Άθροισμα των ακέραιων αριθμών από το 1 ως το 100:

```
INTEGER :: s, i
s = 0
DO i = 1, 100
    s = s + i
END DO
```

- Εκτύπωση των αριθμών 99, 97, 95, ..., 3, 1 (με αυτή τη σειρά):

```
INTEGER :: i
DO i = 99, 1, -2
    PRINT *, i
END DO
```

## Παραδείγματα επανάληψης (2/3)

- Άθροισμα των ακέραιων αριθμών από το 1 ως το 100:

```
INTEGER :: s, i
s = 0
DO i = 1, 100
    s = s + i
END DO
```

- Εκτύπωση των αριθμών 99, 97, 95, ..., 3, 1 (με αυτή τη σειρά):

```
INTEGER :: i
DO i = 99, 1, -2
    PRINT *, i
END DO
```

- Πώς εκτελείται ο κώδικας:

```
INTEGER :: i
DO i = 1, 99, -2
    PRINT *, i
END DO
```



## Παραδείγματα επανάληψης (3/3)

- Εκτύπωση των αριθμών 0.0, 0.1, 0.2, ..., 9.9, 10.0:  
Είναι **ΛΑΘΟΣ** (γιατί;) το

```
DOUBLE PRECISION :: x
DO x = 0.0d0, 10.0d0, 0.1d0
  PRINT *, x
END DO
```

## Παραδείγματα επανάληψης (3/3)

- Εκτύπωση των αριθμών 0.0, 0.1, 0.2, ..., 9.9, 10.0:  
Είναι **ΛΑΘΟΣ** (γιατί;) το

```
DOUBLE PRECISION :: x
DO x = 0.0d0, 10.0d0, 0.1d0
  PRINT *, x
END DO
```

Το **ΣΩΣΤΟ** είναι

```
INTEGER :: i
DO i = 0, 100
  PRINT *, 0.1d0 * i
END DO
```

### Παρατήρηση

Γιατί όχι  $i/10$ ;

## Πότε χρησιμοποιούμε εντολή DO;

Αν έχουμε εντολές που επαναλαμβάνονται και μπορούν να γραφούν στη μορφή

- σύνολο εντολών, εξαρτώμενων ή μη από μια «ακέραια μεταβλητή», αλλά με την ίδια μορφή ανεξάρτητα από την τιμή της μεταβλητής,
- «ακέραια μεταβλητή» = «ακέραια μεταβλητή» + «βήμα αύξησης»,

τότε χρησιμοποιούμε την εντολή DO που παρουσιάσαμε.

# Εφαρμογή: Μετρητής

## Ορισμός

Μια **ακέραια** μεταβλητή που

- **Ακριβώς πριν** το **DO** τη μηδενίζουμε,
- **Μέσα** στις εντολές του **DO** την αυξάνουμε κατά ένα όταν ικανοποιείται κάποια λογική έκφραση,
- **Μετά** το **END DO** έχει τιμή το πόσες φορές στο διάστημα μεταβολής της μεταβλητής ελέγχου ήταν αληθής η λογική έκφραση.

# Εφαρμογή: Μετρητής

## Παράδειγμα

Το πλήθος των ακέραιων στο διάστημα  $[5, 108]$  που είναι πολλαπλάσιοι του 2 ή του 3 υπολογίζεται στο μετρητή  $k$ :

```
INTEGER :: k, i
k = 0
DO i = 5, 108
  IF (MOD(i,2) == 0 .OR. MOD(i,3) == 0) k = k + 1
END DO
PRINT *, k
```

## Εφαρμογή: Λογικός «μετρητής»

Αν ενδιαφερόμαστε όχι για το πόσες φορές αληθεύει μια λογική έκφραση αλλά μόνο για το αν αληθεύει, χρησιμοποιούμε ως «μετρητή» μια μεταβλητή λογικού τύπου. Αμέσως πριν την εντολή **DO** δίνουμε μια τιμή (**.TRUE.** ή **.FALSE.**) και μέσα στο **DO** αλλάζει όταν ικανοποιηθεί κάποια λογική έκφραση.

## Εφαρμογή: Λογικός «μετρητής»

Αν ενδιαφερόμαστε όχι για το **πόσες φορές** αληθεύει μια λογική έκφραση αλλά μόνο για το **αν** αληθεύει, χρησιμοποιούμε ως «μετρητή» μια μεταβλητή λογικού τύπου. **Αμέσως πριν** την εντολή **DO** δίνουμε μια τιμή (**.TRUE.** ή **.FALSE.**) και **μέσα** στο **DO** αλλάζει όταν ικανοποιηθεί κάποια λογική έκφραση.

### Παράδειγμα

Υπάρχει πολλαπλάσιο του 912 στο διάστημα [4000, 5000];

```
LOGICAL :: found
INTEGER :: k
found = .FALSE.
DO k = 4000, 5000
    IF (MOD(k, 912) == 0) found = .TRUE.
END DO
IF (found) PRINT *, "Υπάρχει"
```

# Υπολογισμός αθροίσματος

Κώδικας

Το άθροισμα

$$\sum_{k=i}^j a_k = a_i + a_{i+1} + a_{i+2} + \dots + a_j$$

(με  $i, j$  ακέραια) υπολογίζεται προσθέτοντας σε μια (πραγματική) μεταβλητή έναν-έναν τους (πραγματικούς) όρους:

```
INTEGER :: k
```

```
DOUBLE PRECISION :: s
```

```
s = 0.0d0
```

```
DO k=i, j
```

```
    s = s + ...
```

```
END DO
```

Στο ... γράφουμε τον κώδικα που εκφράζει τον όρο  $a_k$ .

Αφού ολοκληρωθεί το **DO**, η μεταβλητή  $s$  έχει τη ζητούμενη τιμή.



# Υπολογισμός αθροίσματος

Παράδειγμα

Υπολογισμός του αθροίσματος

$$\sum_{k=1}^{10} \frac{1}{k^2} = \frac{1}{1^2} + \frac{1}{2^2} + \dots + \frac{1}{10^2} .$$

```
INTEGER :: k
```

```
DOUBLE PRECISION :: s
```

```
s = 0.0d0
```

```
DO k = 1, 10
```

```
    s = s + 1.0d0/k**2
```

```
END DO
```

# Υπολογισμός αθροίσματος

## Παράδειγμα

Υπολογισμός του αθροίσματος

$$\sum_{k=1}^{10} \frac{1}{k^2} = \frac{1}{1^2} + \frac{1}{2^2} + \dots + \frac{1}{10^2} .$$

```
INTEGER :: k
```

```
DOUBLE PRECISION :: s
```

```
s = 0.0d0
```

```
DO k = 1, 10
```

```
    s = s + 1.0d0/k**2
```

```
END DO
```

## Παρατήρηση

- Γιατί όχι  $1/k**2$ ;

# Υπολογισμός αθροίσματος

## Παράδειγμα

Υπολογισμός του αθροίσματος

$$\sum_{k=1}^{10} \frac{1}{k^2} = \frac{1}{1^2} + \frac{1}{2^2} + \dots + \frac{1}{10^2} .$$

```
INTEGER :: k
DOUBLE PRECISION :: s
s = 0.0d0
DO k = 1, 10
    s = s + 1.0d0/k**2
END DO
```

## Παρατήρηση

- Γιατί όχι `1/k**2`;
- Γιατί όχι `1d0/k*k`;

# Υπολογισμός γινομένου

Κώδικας

Το γινόμενο

$$\prod_{k=i}^j a_k = a_i * a_{i+1} * a_{i+2} * \dots * a_j$$

(με  $i, j$  ακέραια) υπολογίζεται πολλαπλασιάζοντας σε μια πραγματική μεταβλητή έναν-έναν τους όρους:

```
INTEGER :: k
```

```
DOUBLE PRECISION :: p
```

```
p = 1.0d0
```

```
DO k = i, j
```

```
    p = p * ...
```

```
END DO
```

Στο ... γράφουμε τον κώδικα που εκφράζει τον όρο  $a_k$ .

Αφού ολοκληρωθεί το **DO**, η μεταβλητή  $p$  έχει τη ζητούμενη τιμή.

# Υπολογισμός γινομένου

## Παράδειγμα

Το παραγοντικό,  $n!$ , ενός μη αρνητικού ακέραιου  $n$  είναι

$$n! = \begin{cases} 1 \times 2 \times 3 \cdots \times (n-1) \times n = \prod_{k=1}^n k, & n > 0, \\ 1, & n = 0. \end{cases}$$

# Υπολογισμός γινομένου

## Παράδειγμα

Το παραγοντικό,  $n!$ , ενός μη αρνητικού ακέραιου  $n$  είναι

$$n! = \begin{cases} 1 \times 2 \times 3 \cdots \times (n-1) \times n = \prod_{k=1}^n k, & n > 0, \\ 1, & n = 0. \end{cases}$$

Υπολογίζεται στη μεταβλητή  $p$  με τον κώδικα

```
INTEGER :: k, p
p = 1
DO k = 1, n
    p = p * k
END DO
```

Πόσο είναι το  $p$  όταν το  $n$  είναι 0;

# Υπολογισμός διπλού αθροίσματος

Κώδικας

Το άθροισμα

$$\sum_{k=i}^j \sum_{m=p}^n a_{km} = \sum_{k=i}^j \left( \sum_{m=p}^n a_{km} \right)$$

(με  $i, j, p, n$  ακέραια) υπολογίζεται ως εξής:

```
INTEGER :: k
DOUBLE PRECISION :: s, f
s = 0.0d0
DO k = i, j
```

```
    s = s + f
END DO
```

# Υπολογισμός διπλού αθροίσματος

Κώδικας

Το άθροισμα

$$\sum_{k=i}^j \sum_{m=p}^n a_{km} = \sum_{k=i}^j \left( \sum_{m=p}^n a_{km} \right)$$

(με  $i, j, p, n$  ακέραια) υπολογίζεται ως εξής:

```
INTEGER :: k, m
DOUBLE PRECISION :: s, f
s = 0.0d0
DO k = i, j
  f = 0.0d0
  DO m = p, n
    f = f + ...
  END DO
  s = s + f
END DO
```

Στο ... γράφουμε τον κώδικα που εκφράζει τον όρο  $a_{km}$ .



# Υπολογισμός διπλού αθροίσματος

## Παράδειγμα

Πώς υπολογίζεται το

$$\sum_{i=1}^{10} \sum_{j=5}^{15} \frac{1}{i+j} ;$$

# Υπολογισμός διπλού αθροίσματος

## Παράδειγμα

Πώς υπολογίζεται το

$$\sum_{i=1}^{10} \sum_{j=5}^{15} \frac{1}{i+j} ;$$

```
INTEGER :: i, j
DOUBLE PRECISION :: s, g
s = 0.0d0
DO i=1,10
  g = 0d0
  DO j = 5,15
    g = g + 1d0/(i+j)
  END DO
  s = s + g
END DO
```