

Έτοιμα Προγράμματα στον Clipper

/ Το πρόγραμμα αυτό υπολογίζει και τυπώνει το υπόλοιπο της διαίρεσης ενός ακεραίου αριθμού με το τρία */*

```
CLEAR SCREEN
NUM = 0
@ 5, 6 SAY 'Δώσε έναν ακέραιο αριθμό : ' GET NUM PICTURE '9999'
READ
YPOL = NUM % 3

DO CASE
    CASE YPOL = 0
        ? 'υπόλοιπο = 0'
    CASE YPOL = 1
        ? 'υπόλοιπο = 1'
    CASE YPOL = 2
        ? 'υπόλοιπο = 2'
    OTHERWISE
        ? 'αδύνατη περίπτωση'
ENDCASE
QUIT
```

/ Το πρόγραμμα αυτό υπολογίζει και τυπώνει το άθροισμα κάποιων αριθμών· τελειώνει δε όταν το άθροισμά τους ξεπεράσει το 1.000 */*

```
CLEAR SCREEN
NUM = 0
SUM = 0

DO WHILE SUM <= 1000
    @ 5, 6 SAY 'Δώσε έναν ακέραιο αριθμό : ' GET NUM;
        PICTURE '9999'
    READ
    SUM = SUM + NUM
ENDDO

? 'Το άθροισμα των αριθμών είναι : ' + SUM
QUIT
```

/ Το πρόγραμμα αυτό υπολογίζει τον φόρο ενός μισθωτού για ένα χρόνο δίνοντας σαν δεδομένο το ετήσιο εισόδημά του και τον αριθμό των παιδιών του. Ο φόρος υπολογίζεται σαν το ένα τρίτο του εισοδήματος αφού αφαιρέσουμε τις απαλλαγές, που είναι η ατομική έκπτωση (500.000) καθώς και η έκπτωση για το κάθε παιδί (100.000) */*

CLEAR SCREEN

ATOM_EKPT = 500000

EKPT_CHILD = 100000

EISODHMA = 0

CHILD = 0

@ 5, 6 SAY 'Δώσε το εισόδημα : ' GET EISODHMA PICTURE '9999999'

@ 6, 6 SAY 'Δώσε τα παιδιά : ' GET CHILD PICTURE '99'

READ

EISODHMA = EISODHMA - ATOM_EKPT - EKPT_CHILD*CHILD

FOROS = EISODHMA / 3

? 'Ο φόρος είναι : ' + FOROS

QUIT

/ Το πρόγραμμα αυτό υπολογίζει τον λογαριασμό της ΔΕΗ ενός καταναλωτή με δεδομένο ότι για τις πρώτες 100 μονάδες η 1 KWh χρεώνεται 15 δραχ., για τις επόμενες μέχρι τις 200 χρεώνεται 16 δραχ. και από 201 και πάνω μονάδες χρεώνεται 17 δραχ. Ο ΦΠΑ είναι 18%, το ποσό για την ΕΡΤ είναι 5% και το πάγιο είναι 1.500 δραχ. */*

CLEAR SCREEN

KATAN = 0

PAGIO = 1500

@ 5, 6 SAY 'Δώσε την κατανάλωση : ' GET KATAN PICTURE '9999'

READ

IF KATAN <= 100

LOGAR = KATAN * 15

ENDIF

IF KATAN > 100 .AND. KATAN <= 200

LOGAR = 100*15 + (KATAN-100) * 16

ENDIF

IF KATAN > 200

LOGAR = 100*15 + 100*16 + (KATAN-200)*17

ENDIF

LOGAR = LOGAR + PAGIO

ERT = ROUND(LOGAR * 0.05, 0)

LOGAR = LOGAR + ERT

FPA = ROUND(LOGAR * 0.18, 0)

LOGAR = LOGAR + FPA

QUIT

/ Το πρόγραμμα αυτό βρίσκει τον μεγαλύτερο από τρεις αριθμούς A, B, C. Χρειάζεται να ορίσουμε μια βοηθητική μεταβλητή MAX όπου εκεί θα καταχωρούμε προσωρινά τον μεγαλύτερο αριθμό */*

```
CLEAR SCREEN
```

```
A = 0
```

```
B = 0
```

```
C = 0
```

```
@ 5, 6 SAY 'Δώσε τον 1ο αριθμό : ' GET A PICTURE '999'
```

```
@ 6, 6 SAY 'Δώσε τον 2ο αριθμό : ' GET B PICTURE '999'
```

```
@ 7, 6 SAY 'Δώσε τον 3ο αριθμό : ' GET C PICTURE '999'
```

```
READ
```

```
MAX = A
```

```
IF B > MAX
```

```
    MAX = B
```

```
ENDIF
```

```
IF C > MAX
```

```
    MAX = C
```

```
ENDIF
```

```
? 'Μεγαλύτερος είναι ο : ' + MAX
```

```
QUIT
```

*/*Το πρόγραμμα αυτό βρίσκει αν ένας ακέραιος αριθμός είναι μονός ή ζυγός*/*

```
CLEAR SCREEN
```

```
NUM = 0
```

```
@ 5, 6 SAY 'Δώσε έναν αριθμό : ' GET NUM PICTURE '999'
```

```
READ
```

```
IF (NUM % 2) = 0
```

```
    ? 'Ο αριθμός είναι ζυγός'
```

```
ELSE
```

```
    ? 'Ο αριθμός είναι μονός'
```

```
ENDIF
```

```
QUIT
```

/ Το πρόγραμμα αυτό υπολογίζει τον μισθό ενός εργάτη όταν δίνονται οι μέρες εργασίας του, οι υπερωρίες του σε ώρες και είναι γνωστό ότι οι κρατήσεις είναι το 20% των αποδοχών του, οι υπερωρίες του πληρώνονται με το 20% του ημερομισθίου και ο φόρος είναι 5% */*

```
CLEAR SCREEN
MERES_ERG = 0
HMEROMISTH = 0
YPER = 0
```

```
@ 5, 6 SAY 'Μέρες εργασίας : ' GET MERES_ERG PICTURE '99'
@ 6, 6 SAY 'Ημερομίσθιο : ' GET HMEROMISTH PICTURE '99999'
@ 7, 6 SAY 'Υπερωρίες : ' GET YPER PICTURE '999'
READ
MIKTA = ROUND(HMEROMISTH * MERES_ERG +
              YPER * 0.2 * HMEROMISTH, 0)
KRATHSEIS = ROUND(MIKTA * 0.2, 0)
FOROS = ROUND((MIKTA-KRATHSEIS) * 0.05, 0)
KATHARA = MIKTA - KRATHSEIS - FOROS
```

```
QUIT
```

/ Το πρόγραμμα αυτό διαβάζει έναν διψήφιο ακέραιο αριθμό και υπολογίζει και εκτυπώνει τον ακέραιο που προκύπτει από την αντιστροφή των ψηφίων του */*

```
CLEAR SCREEN
I = 0
```

```
DO WHILE .T.
  @ 6, 6 SAY 'Δώσε έναν ακέραιο : ' GET I PICTURE '99'
  READ
  IF (I > 0 .AND. I < 100)
    EXIT
  ELSE
    LOOP
  ENDIF
```

```
ENDDO
```

/ ο παραπάνω βρόχος διαβάζει έναν ακέραιο και αν αυτός είναι διψήφιος, βγαίνουμε από τον βρόχο με την εντολή EXIT και αν όχι, επιστρέφουμε στην αρχή του βρόχου με την εντολή LOOP */*

```
PILIKO = I / 10          /* απομονώνει το πρώτο ψηφίο του αριθμού */
YPOL = I % 10          /* απομονώνει το δεύτερο ψηφίο του αριθμού */
NEOS_I = YPOL * 10 + PILIKO
? 'Ο νέος ακέραιος είναι : ' + NEOS_I
QUIT
```

/ Το πρόγραμμα αυτό στρογγυλοποιεί έναν μισθό στο τάληρο, δηλ. αν λήγει σε 0, 1 και 2 τον κάνει να λήγει σε 0, αν λήγει σε 3, 4, 5, 6 ή 7 τον κάνει να λήγει σε 5 και αν λήγει σε 8 ή 9 τον κάνει να λήγει σε 0 αλλά αυξημένο κατά 10 */*

```
CLEAR SCREEN
MISTHOS = 0
```

```
@ 5, 6 SAY 'Δώσε τον μισθό : ' GET MISTHOS PICTURE '999999'
READ
```

```
PILIKO = MISTHOS / 10
YPOL = MISTHOS % 10  /* απομονώνει το τελευταίο ψηφίο του μισθού */
```

```
DO CASE
  CASE YPOL < 3
    MISTHOS = PILIKO * 10
  CASE YPOL > 2 .AND. YPOL < 8
    MISTHOS = PILIKO * 10 + 5
  CASE YPOL > 7
    MISTHOS = PILIKO * 10 + 10
ENDCASE
```

```
? 'Ο νέος μισθός είναι : ' + MISTHOS
QUIT
```

/ Το πρόγραμμα αυτό εκτυπώνει τους ακέραιους από το 1 μέχρι το 20 */*

```
CLEAR SCREEN

FOR I := 1 TO 20
  ? I
NEXT I
```

```
QUIT
```

```
ή
```

```
CLEAR SCREEN

I := 1
DO WHILE I <= 20
  ? I
  I = I + 1
ENDDO
```

/ Το πρόγραμμα αυτό υπολογίζει το άθροισμα και το γινόμενο N αριθμών */*

CLEAR SCREEN

N = 0

A = 0

SUM = 0

PRODUCT = 1

@ 5, 6 SAY 'Δώσε το πλήθος των αριθμών : ' GET N PICTURE '999'
READ

FOR I:= 1 TO N

 @ 6, 6 SAY 'Δώσε έναν αριθμό : ' GET A PICTURE '99999'

 READ

 SUM = SUM + A

 PRODUCT = PRODUCT * A

NEXT I

? 'Το άθροισμα των αριθμών είναι : ' + SUM

? 'Το γινόμενο των αριθμών είναι : ' + PRODUCT

QUIT

/ Το πρόγραμμα αυτό βρίσκει τον μεγαλύτερο από N αριθμούς */*

CLEAR SCREEN

N = 0

A = 0

@ 5, 6 SAY 'Δώσε το πλήθος των αριθμών : ' GET N PICTURE '999'
READ

@ 6, 6 SAY 'Δώσε έναν αριθμό : ' GET A PICTURE '99999'

READ

MAX = A

FOR I:= 2 TO N

 @ 6, 6 SAY 'Δώσε έναν αριθμό : ' GET A PICTURE '99999'

 READ

 IF A > MAX

 MAX = A

 ENDIF

NEXT I

? 'Ο μεγαλύτερος αριθμός είναι ο : ' + MAX

QUIT

/ Το πρόγραμμα αυτό βρίσκει πόσοι από N αριθμούς είναι θετικοί, πόσοι αρνητικοί και πόσοι μηδέν */*

CLEAR SCREEN

N = 0

A = 0

POS = 0

NEG = 0

ZERO = 0

@ 5, 6 SAY 'Δώσε το πλήθος των αριθμών : ' GET N PICTURE '99'
READ

FOR I:= 1 TO N

 @ 6, 6 SAY 'Δώσε έναν αριθμό : ' GET A PICTURE '99999'

 READ

 IF A > 0

 POS = POS + 1

 ENDIF

 IF A < 0

 NEG = NEG + 1

 ENDIF

 IF A = 0

 ZERO = ZERO + 1

 ENDIF

NEXT I

? 'Υπάρχουν ' + POS + ' θετικοί αριθμοί'

? 'Υπάρχουν ' + NEG + ' αρνητικοί αριθμοί'

? 'Υπάρχουν ' + ZERO + ' αριθμοί ίσοι με μηδέν'

QUIT

/ Το πρόγραμμα αυτό υπολογίζει το σύνολο των μαθητών και τον μέσο όρο βαθμολογίας για κάθε τάξη, για τρεις τάξεις A, B, C και για N μαθητές */*

CLEAR SCREEN

N = 0

SUM_A = 0

SUM_B = 0

SUM_C = 0

SUM_B_A = 0

SUM_B_B = 0

SUM_B_C = 0

@ 5, 6 SAY 'Δώσε το πλήθος των μαθητών : ' GET N PICTURE '99'
READ

FOR I := 1 TO N

@ 6, 6 SAY 'Δώσε την τάξη : ' GET TAXI PICTURE 'A'
VALID TAXI \$ 'ABC'

@ 7, 6 SAY 'Δώσε τον βαθμό : ' GET B PICTURE '99'
RANGE 0, 20

READ

DO CASE

CASE TAXI == 'A'

SUM_A = SUM_A + 1

SUM_B_A = SUM_B_A + B

CASE TAXI == 'B'

SUM_B = SUM_B + 1

SUM_B_B = SUM_B_B + B

CASE TAXI == 'C'

SUM_C = SUM_C + 1

SUM_B_C = SUM_B_C + B

OTHERWISE

? 'Προσοχή! Λάθος!'

ENDCASE

NEXT I

?? 'Η πρώτη τάξη έχει ' + SUM_A + ' μαθητές και μέσο όρο βαθμολογίας '
? ROUND(SUM_B_A/SUM_A, 1)

?? 'Η δεύτερα τάξη έχει ' + SUM_B + ' μαθητές και μέσο όρο βαθμολογίας '
? ROUND(SUM_B_B/SUM_B, 1)

?? 'Η τρίτη τάξη έχει ' + SUM_C + ' μαθητές και μέσο όρο βαθμολογίας '
? ROUND(SUM_B_C/SUM_C, 1)

/ Το πρόγραμμα αυτό διαβάζει έναν τριψήφιο ακέραιο αριθμό και υπολογίζει και εκτυπώνει τον ακέραιο που προκύπτει από την αντιστροφή των ψηφίων του */*

```
CLEAR SCREEN
I = 0
DO WHILE .T.
    @ 6, 6 SAY 'Δώσε έναν τριψήφιο ακέραιο : ' GET I PICTURE '999'
    READ
    IF I > 99 .AND. I < 1000
        EXIT
    ELSE
        LOOP
    ENDIF
ENDDO
```

/ ο παραπάνω βρόχος διαβάζει έναν ακέραιο και αν αυτός είναι τριψήφιος, βγαίνουμε από τον βρόχο με την εντολή EXIT και αν όχι, επιστρέφουμε στην αρχή του βρόχου με την εντολή LOOP */*

```
PILIKO1 = I / 100          /* απομονώνει το 1ο ψηφίο του αριθμού */
YPOL1 = I % 100          /* απομονώνει τα 2 δεξιά ψηφία του αριθμού */
PILIKO2 = YPOL1 / 10     /* απομονώνει το 2ο ψηφίο του αριθμού */
YPOL2 = YPOL1 % 10     /* απομονώνει το 3ο ψηφίο του αριθμού */
NEOS_I = YPOL2 * 100 + PILIKO2 * 10 + PILIKO1
? 'Ο νέος ακέραιος είναι : ' + NEOS_I
QUIT
```

/ Δίνονται τα παρακάτω στοιχεία για 20 ανθρώπους :*

- όνομα

- κωδικός φύλου

*Ζητάμε να βρεθούν πόσες είναι οι γυναίκες και πόσοι είναι οι άνδρες */*

```
CLEAR SCREEN
DECLARE NAME[20], FYLO[20]
ANDRES := 0
GYNAIKES := 0
FOR I := 1 TO 20
    NAME[I] := SPACE(15)
    FYLO[I] := .T.
NEXT I
FOR I := 1 TO 20
    @ 3, 5 SAY 'Δώσε το όνομα : ' GET NAME[I] PICTURE '@A'
    @ 4, 5 SAY 'Δώσε το φύλο (Y/N) : ' GET FYLO[I] PICTURE 'Y'
    READ
    IF FYLO[I]
        ANDRES := ANDRES + 1
    ELSE
        GYNAIKES := GYNAIKES + 1
    ENDIF
NEXT I
QUIT
```

/ Δίνονται τα εξής στοιχεία για 20 μαθητές μιας τάξης :*

α. όνομα

β. κωδικός φύλου

γ. βαθμός

*Να βρεθεί ποιο αγόρι έχει τον μικρότερο βαθμό στην τάξη και ποιο κορίτσι τον μεγαλύτερο */*

CLEAR SCREEN

DECLARE NAME[20], FYLO[20], VATHMOS[20]

MIN := 21

MAX := 0

FOR I := 1 TO 20

 NAME[I] := SPACE(15)

 FYLO[I] := .T.

 VATHMOS[I] := 0

NEXT I

FOR I := 1 TO 20

 @ 3, 5 SAY 'Δώσε το όνομα : ' GET NAME[I] PICTURE '@A'

 @ 4, 5 SAY 'Δώσε το φύλο (Y/N) : ' GET FYLO[I] PICTURE 'Y'

 @ 3, 5 SAY 'Δώσε τον βαθμό : ' GET VATHMOS[I] PICTURE '99'

 RANGE 0, 20

 READ

 IF FYLO[I] = .T. .AND. VATHMOS[I] < MIN

 MIN := VATHMOS[I]

 MIN_NAME := NAME[I]

 ENDIF

 IF FYLO[I] = .F. .AND. VATHMOS[I] > MAX

 MAX := VATHMOS[I]

 MAX_NAME := NAME[I]

 ENDIF

NEXT I

? 'Το αγόρι με τον μικρότερο βαθμό στην τάξη είναι ο : ' + MIN_NAME

? 'Το κορίτσι με τον μεγαλύτερο βαθμό στην τάξη είναι η : ' + MAX_NAME

QUIT

/ Μια εταιρεία μετράει τη μόλυνση της ατμόσφαιρας και καταχωρεί τις ενδείξεις της σε ακέραιες τιμές από το 0 έως το 100 με μια μέτρηση για κάθε μέρα. Να γίνει πρόγραμμα που να καταχωρεί τις τιμές της ατμοσφαιρικής μόλυνσης για 20 μέρες και να βρίσκει τις κορυφές της μόλυνσης, δηλ. αυτές που είναι μεγαλύτερες από τη μόλυνση της προηγούμενης και της επόμενης μέρας. Για κάθε κορυφή το πρόγραμμα θα τυπώνει τη μέρα που έγινε και τη μόλυνση που υπήρχε εκείνη τη μέρα */*

```
CLEAR SCREEN
```

```
DECLARE POLLUTION[20]
```

```
FOR I:= 1 TO 20
```

```
    POLLUTION[I] := 0
```

```
NEXT I
```

```
FOR I:= 1 TO 20
```

```
    @ 3, 5 SAY 'Δώσε τη μόλυνση : ' GET POLLUTION[I];
```

```
    PICTURE '99' RANGE 0, 100
```

```
    READ
```

```
NEXT I
```

```
FOR I:= 2 TO 19
```

```
    IF POLLUTION[I] > POLLUTION[I-1] .AND.;
```

```
        POLLUTION[I] > POLLUTION[I+1]
```

```
        ? 'Υπάρχει κορυφή την '+I+'η μέρα με τιμή '+POLLUTION[I]
```

```
    ENDIF
```

```
NEXT I
```

```
QUIT
```

/ Να ταξινομηθεί κατ' αύξουσα σειρά πίνακας A με 20 στοιχεία. Θα εφαρμοσθεί η μέθοδος Bubble-Sort, σύμφωνα με την οποία σαρώνουμε συνέχεια τον πίνακα και αν κάποια στοιχεία δεν είναι στη σωστή σειρά, τότε τα κάνουμε ανταλλαγή και βάζουμε την τιμή 1 σε μια σημαία flag. Όταν σαρώσουμε τον πίνακα και η flag είναι ίση με μηδέν, τότε αυτό θα σημαίνει ότι ο πίνακας είναι ταξινομημένος */*

```
CLEAR SCREEN
DECLARE A[20]
FLAG := 1

FOR I := 1 TO 20
    A[I] := 0
NEXT I

FOR I := 1 TO 20
    @ 3,5 SAY 'Δώσε έναν αριθμό : ' GET A[I] PICTURE '999'
    READ
NEXT I

DO WHILE .T.
    FLAG = 0
    FOR I := 1 TO 19
        IF A[I] > A[I+1]
            TEMP := A[I]
            A[I] := A[I+1]
            A[I+1] := TEMP
            FLAG := 1
        ENDIF
    NEXT I
    IF FLAG = 0
        EXIT
    ENDIF
ENDDO

QUIT
```

/ Δίνονται τα εξής στοιχεία για 20 νέους οπλίτες που πρόκειται να καταταγούν στα διάφορα όπλα του στρατού :*

- ονοματεπώνυμο

- όπλο κατάταξης (1=στρατός ξηράς, 2=ναυτικό, 3=αεροπορία)

*Να γίνει εκτύπωση για τα τρία όπλα ξεχωριστά */*

```

CLEAR SCREEN
DECLARE NAME[20], OPLO[20], STRATOS[20], NAYTIKO[20],
        AEROPORIA[20]
I_STRATOS := 0
I_NAYTIKO := 0
I_AEROPORIA := 0
FOR I:= 1 TO 20
    NAME[I] := SPACE(15)
    OPLO[I] := 0
NEXT I
FOR I:= 1 TO 20
    @ 3,5 SAY 'Δώσε το όνομα : ' GET NAME[I] PICTURE '@A'
    @ 4,5 SAY 'Δώσε το όπλο (1,2,3) : ' GET OPLO[I] PICTURE '9'
    RANGE 1,3
    READ
    IF OPLO[I] = 1
        I_STRATOS := I_STRATOS + 1
        STRATOS[I_STRATOS] := NAME[I]
    ELSEIF OPLO[I] = 2
        I_NAYTIKO := I_NAYTIKO + 1
        NAYTIKO[I_NAYTIKO] := NAME[I]
    ELSE
        I_AEROPORIA := I_AEROPORIA + 1
        AEROPORIA[I_AEROPORIA] := NAME[I]
    ENDIF
NEXT I

? 'Στον στρατό ξηράς υπηρετούν οι : '
FOR I:= 1 TO I_STRATOS
    ? STRATOS[I]
NEXT I

? 'Στο ναυτικό υπηρετούν οι : '
FOR I:= 1 TO I_NAYTIKO
    ? NAYTIKO[I]
NEXT I

? 'Στην αεροπορία υπηρετούν οι : '
FOR I:= 1 TO I_AEROPORIA
    ? AEROPORIA[I]
NEXT I
QUIT

```