



ΑΣΚΗΣΕΙΣ - ΕΠΑΝΑΛΗΨΗ

ΕΠΙΛΥΣΗ ΑΣΚΗΣΗΣ ΓΙΑ ΣΠΙΤΙ

ΑΣΚΗΣΗ ΓΙΑ ΣΠΙΤΙ

Κάθε εκτυπωτής χρησιμοποιεί μια ουρά εκτύπωσης για να τοποθετεί σε αυτήν τα αρχεία που λαμβάνει για εκτύπωση με τη σειρά με την οποία στάλθηκαν. Κάθε φορά εκτυπώνεται το αρχείο που βρίσκεται στην αρχή της ουράς εκτύπωσης. Λόγω της περιορισμένης μνήμης του εκτυπωτή θεωρούμε ότι στην ουρά μπορούν να εισαχθούν το πολύ 20 αρχεία. Να αναπτύξετε πρόγραμμα το οποίο :

- 1) Διαβάζει την επιλογή του χρήστη, που θα πρέπει να είναι “N” ή “P” ή “E”. Αν επιλεγθεί το “N” θα ζητάει το πρόγραμμα από το χρήστη το όνομα του αρχείου και θα το εισάγει στην ουρά εκτύπωσης(εφόσον υπάρχει χώρος), εμφανίζοντας κατάλληλο μήνυμα. Σε περίπτωση που δεν υπάρχει χώρος τότε θα εμφανίζει αντίστοιχα κάποιο κατάλληλο μήνυμα.
- 2) Αν ο χρήστης δώσει την επιλογή “P” θα εξετάζεται αν υπάρχει αρχείο προς εκτύπωση και εφόσον υπάρχει θα εμφανίζεται το μήνυμα “Printed” μαζί με το όνομα του αρχείου. Αν δεν υπάρχει αρχείο προς εκτύπωση, τότε θα εμφανίζεται ένα κατάλληλο μήνυμα
- 3) Η διαδικασία αυτή θα ολοκληρώνεται αν ο χρήστης δώσει την επιλογή “E” ή αν αδειάσει η ουρά εκτύπωσης και αφού συμβεί αυτό, θα εκτυπώνονται στην οθόνη ο αριθμός των αρχείων που εκτυπώθηκαν καθώς και ο αριθμός των φορών που η ουρά εκτύπωσης είχε μόνο 1 αρχείο

ΘΕΩΡΗΣΤΕ ΟΤΙ ΣΤΑΛΘΗΚΕ ΤΟΥΛΑΧΙΣΤΟΝ 1 ΑΡΧΕΙΟ

ΕΠΙΛΥΣΗ

Η εκφώνηση της άσκησης μας ενημερώνει πως στάλθηκε τουλάχιστον 1 αρχείο καθώς την πρώτη φορά που φορτώνεται το πρόγραμμα η ουρά εκτύπωσης είναι άδεια και στο μενού επιλογών του χρήστη αν αυτός επιλέξει “P” τότε το πρόγραμμα θα τερματίζει καθώς αρχικά δεν υπάρχει αρχείο στην ουρά εκτύπωσης. Πρακτικά η εκφώνηση αναφέρει πως την πρώτη φορά ο χρήστης εννοείται πως θα πατήσει “N” για εισαγωγή στην ουρά εκτύπωσης ενός αρχείου

```
ΠΡΟΓΡΑΜΜΑ ΦΟΡΟΛΟΓΟΥΜΕΝΟΙ_ΟΥΡΑ
ΣΤΑΘΕΡΕΣ
    N = 20
ΜΕΤΑΒΛΗΤΕΣ
    ΧΑΡΑΚΤΗΡΕΣ: user_input, QUEUE[N], filename
    ΑΚΕΡΑΙΕΣ: front, rear, counter_printed, counter_singleFile
ΑΡΧΗ
    ΓΡΑΨΕ 'Αρχειοποίηση ουράς εκτυπωτή'
    front <- 0
    rear <- 0
    counter_singleFile <- 0
    counter_printed <- 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΓΡΑΨΕ 'Τι θέλεις να κάνεις...'
        ΓΡΑΨΕ 'N για λήψη νέου αρχείου'
        ΓΡΑΨΕ 'P για εκτύπωση πρόσφατου αρχείου'
        ΔΙΑΒΑΣΕ user_input
        ΜΕΧΡΙΣ_ΟΤΟΥ user_input = 'N' Η user_input = 'P' Η user_input = 'E'

        ΑΝ user_input = 'N' ΤΟΤΕ
            ΑΝ rear = N ΤΟΤΕ
                ΓΡΑΨΕ 'Δεν υπάρχει χώρος στην ουρά'
            ΑΛΛΙΩΣ
                ΓΡΑΨΕ 'Δώσε το όνομα του αρχείου για εισαγωγή'
                ΔΙΑΒΑΣΕ filename
                ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
                    ΓΡΑΨΕ 'Εισαγωγή πρώτου αρχείου στην ουρά'
                    front <- 1
                ΤΕΛΟΣ_ΑΝ
                rear <- rear + 1
                QUEUE[rear] <- filename
                ΓΡΑΨΕ 'Η εισαγωγή του αρχείου ολοκληρώθηκε!'
            ΤΕΛΟΣ_ΑΝ
            ΑΛΛΙΩΣ_ΑΝ user_input = 'P' ΤΟΤΕ
                ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
                    ΓΡΑΨΕ 'Δεν υπάρχει αρχείο για εκτύπωση'
                ΑΛΛΙΩΣ
                    ΓΡΑΨΕ 'Filename ', QUEUE[front], ' printed'
                    counter_printed <- counter_printed + 1
                    ΑΝ front = rear ΤΟΤΕ
                        ΓΡΑΨΕ 'Η ουρά εκτύπωσης μόλις άδειασε'
                        front <- 0
                        rear <- 0
                    ΑΛΛΙΩΣ
                        front <- front + 1
                    ΤΕΛΟΣ_ΑΝ
                ΤΕΛΟΣ_ΑΝ
            ΑΛΛΙΩΣ_ΑΝ user_input = 'E' ΤΟΤΕ
                ΓΡΑΨΕ 'Σε ευχαριστούμε!'
            ΑΛΛΙΩΣ
                ΓΡΑΨΕ 'Δώσε N, P ή E...'
            ΤΕΛΟΣ_ΑΝ
            ΑΝ (front = rear) ΚΑΙ front <> 0 ΤΟΤΕ
                counter_singleFile <- counter_singleFile + 1
            ΤΕΛΟΣ_ΑΝ
        ΜΕΧΡΙΣ_ΟΤΟΥ (user_input = 'E') Η (front = 0 ΚΑΙ rear = 0)

        ΓΡΑΨΕ 'Συνολικά εκτυπώθηκαν ', counter_printed, ' αρχεία'
        ΓΡΑΨΕ 'Η ουρά εκτύπωσης περιείχε 1 αρχείο ', counter_singleFile, ' φορές'
    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΑΣΚΗΣΗ 1 ΜΕ ΠΙΝΑΚΑ ΣΥΧΝΟΤΗΤΩΝ

ΝΓΠ το οποίο να δέχεται τους βαθμούς μιας τάξης 15 μαθητών στο μάθημα της φυσικής και στη συνέχεια να εμφανίζει το βαθμό που παρατηρήθηκε τις περισσότερες φορές (επικρατούσα τιμή), όπως και πόσες φορές εμφανίστηκε.

Θεωρήστε ότι οι βαθμοί είναι ακέραιοι και ανήκουν στο διάστημα 1-20. Στο τέλος να παρουσιάζει μόνο τους διαφορετικούς βαθμούς που εμφανίστηκαν στην τάξη κατά αύξουσα σειρά

```

1 ΠΡΟΓΡΑΜΜΑ ΒΑΘΜΟΛΟΓΙΕΣ
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: i, j, βαθμός, ΒΑΘΜΟΙ[15], ΣΥΧΝΟΤΗΤΕΣ[20], index_βοηθητικού, max_συχνότητα, max_βαθμός, ΒΟΗΘ[20], temp
4   ΑΡΧΗ
5     ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
6       ΓΡΑΨΕ 'Δώσε βαθμολογία μαθητή'
7       ΔΙΑΒΑΣΕ βαθμός
8       ΟΣΟ (βαθμός <= 0 Η βαθμός > 20) ΕΠΑΝΑΛΑΒΕ
9         ΓΡΑΨΕ 'Δώσε έγκυρο βαθμό'
10        ΔΙΑΒΑΣΕ βαθμός
11      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
12      ΒΑΘΜΟΙ[i] <- βαθμός
13    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
14
15    ΓΡΑΨΕ 'Η καταχώρηση βαθμολογιών ολοκληρώθηκε'
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

```



```
37 index_βοηθητικού <- 0
38 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
39     ΑΝ ΣΥΧΝΟΤΗΤΕΣ[i] > 0 ΤΟΤΕ
40         index_βοηθητικού <- index_βοηθητικού + 1
41         ΒΟΗΘ[index_βοηθητικού] <- i
42     ΤΕΛΟΣ_ΑΝ
43 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
44
```

```
44
45 ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ index_βοηθητικού
46     ΓΙΑ j ΑΠΟ index_βοηθητικού ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
47         ΑΝ ΒΟΗΘ[j] < ΒΟΗΘ[j-1] ΤΟΤΕ
48             temp <- ΒΟΗΘ[j]
49             ΒΟΗΘ[j] <- ΒΟΗΘ[j-1]
50             ΒΟΗΘ[j-1] <- temp
51         ΤΕΛΟΣ_ΑΝ
52     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
53 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
54
```

```
54
55 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ index_βοηθητικού
56     ΓΡΑΨΕ ΒΟΗΘ[i]
57 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
58
59 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
60
```

ΑΣΚΗΣΗ 2

- ΚΛΙΜΑΚΩΤΗ ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ
- ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΜΟΝΟΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ
- ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΑΛΓΟΡΙΘΜΟ ΕΥΘΕΙΑΣ ΑΝΤΑΛΛΑΓΗΣ

Το τμήμα μισθοδοσίας μιας εταιρείας λαμβάνει από το τμήμα προσωπικού τα στοιχεία που αφορούν την απασχόληση των εργαζομένων για τον προηγούμενο μήνα: πρόκειται για τον πίνακα ΟΝ, που περιέχει τα ονόματα των 1200 εργαζομένων και τον πίνακα ΩΡΕΣ, που περιέχει τις ώρες που εργάστηκαν. Ο μισθός κάθε εργαζομένου προκύπτει σύμφωνα με τον παρακάτω πίνακα (κλιμακωτός υπολογισμός)

Ώρες εργασίας	Ωρομίσθιο
1-25	20
26-35	25
36 και άνω	30

ΝΑΠ που θα δημιουργεί τον πίνακα μισθοδοσίας και στην συνέχεια:

- Θα εκτυπώνει για όλους τους εργαζομένους όνομα και μισθό
- Θα διαβάζει το όνομα ενός εργαζομένου και θα εκτυπώνει τον μισθό του
- Θα πρέπει να δοθεί αύξηση 5% στους 50 πιο χαμηλόμισθους. Το πρόγραμμα θα πρέπει να εκτυπώνει τα ονόματα και τον νέο μισθό αυτών των υπαλλήλων (είναι δεδομένο ότι οι δικαιούχοι είναι τελικά μόνο 50)

```

1 ΠΡΟΓΡΑΜΜΑ ΑΣΚΗΣΗ3
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: i, j, pos
4   ΛΟΓΙΚΕΣ: done
5   ΧΑΡΑΚΤΗΡΕΣ: ON[1200], ΟΝΟΜΑ, temp2
6   ΠΡΑΓΜΑΤΙΚΕΣ: ΜΙΣΘ[1200], ΩΡΕΣ[1200], temp1
7 ΑΡΧΗ
8   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 1200
9       ΓΡΑΨΕ 'Δώσε όνομα'
10      ΔΙΑΒΑΣΕ ON[i]
11      ΓΡΑΨΕ 'Δώσε ώρες εργασίας'
12      ΔΙΑΒΑΣΕ ΩΡΕΣ[i]
13  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
14

```

```

15  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 1200
16      ΑΝ ΩΡΕΣ[i] <= 25 ΤΟΤΕ
17          ΜΙΣΘ[i] <- ΩΡΕΣ[i] * 20
18      ΑΛΛΙΩΣ_ΑΝ ΩΡΕΣ[i] <= 35 ΤΟΤΕ
19          ΜΙΣΘ[i] <- (25 * 20) + (ΩΡΕΣ[i] - 25) * 25
20      ΑΛΛΙΩΣ
21          ΜΙΣΘ[i] <- (25 * 20) + (10 * 25) + (ΩΡΕΣ[i] - 35) * 30
22      ΤΕΛΟΣ_ΑΝ
23  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
24
25  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 1200
26      ΓΡΑΨΕ 'Όνομα: ', ON[i], ' με εισόδημα: ', ΜΙΣΘ[i]
27  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
28

```



```
30 ΓΡΑΨΕ 'Δώσε όνομα υπάλληλου...'  
31 ΔΙΑΒΑΣΕ ΟΝΟΜΑ  
32 done <- ΨΕΥΔΗΣ  
33 pos <- 0  
34 i <- 1  
35 ΟΣΟ(i <= 1200 ΚΑΙ done=ΨΕΥΔΗΣ) ΕΠΑΝΑΛΑΒΕ  
36     ΑΝ ΟΝ[i] = ΟΝΟΜΑ ΤΟΤΕ  
37         done <- ΑΛΗΘΗΣ  
38         pos <- i  
39     ΑΛΛΙΩΣ  
40         i <- i + 1  
41     ΤΕΛΟΣ_ΑΝ  
42 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ  
43  
44 ΑΝ pos <> 0 ΤΟΤΕ  
45     ΓΡΑΨΕ 'Ο υπάλληλος ', ΟΝΟΜΑ, ' παίρνει μισθό: ', ΜΙΣΘ[pos]  
46 ΑΛΛΙΩΣ  
47     ΓΡΑΨΕ 'Δεν βρέθηκε υπάλληλος με αυτό το όνομα'  
48 ΤΕΛΟΣ_ΑΝ  
49
```

```

69 ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 51
70   ΓΙΑ j ΑΠΟ 1200 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
71     ΑΝ ΜΙΣΘ[j-1] > ΜΙΣΘ[j] ΤΟΤΕ
72       temp1 <- ΜΙΣΘ[j-1]
73       ΜΙΣΘ[j-1] <- ΜΙΣΘ[j]
74       ΜΙΣΘ[j] <- temp1
75
76       temp1 <- ΩΡΕΣ[j-1]
77       ΩΡΕΣ[j-1] <- ΩΡΕΣ[j]
78       ΩΡΕΣ[j] <- temp1
79
80       temp2 <- ΟΝ[j-1]
81       ΟΝ[j-1] <- ΟΝ[j]
82       ΟΝ[j] <- temp2
83
84     ΤΕΛΟΣ_ΑΝ
85   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
86 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
87
88
89 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
90   ΓΡΑΨΕ 'Ο νέος μισθός του υπαλλήλου ', ΟΝ[i], ' είναι ', ΜΙΣΘ[i] + ΜΙΣΘ[i] * 0.05
91 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
92
93 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
94

```

ΑΣΚΗΣΗ 3

Αλυσίδα δισκοπολείων με 15 υποκαταστήματα καταγράφει τα CD που διαθέτει. ΝΑΠ το οποίο:

- **ΑΝΤΙΜΕΤΑΘΕΣΗ ΠΙΝΑΚΩΝ – ΠΑΡΑΛΛΗΛΟΙ ΠΙΝΑΚΕΣ**
- **ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΚΑΤΑ ΓΡΑΜΜΕΣ ΚΑΙ ΣΤΗΛΕΣ ΣΕ ΔΙΣΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ**
- **ΕΥΡΕΣΗ ΠΛΗΘΟΥΣ**
- Θα διαβάζει τα ονόματα των 15 δισκοπολείων στον 1D $\Delta[15]$, τους 1000 διαφορετικούς τίτλους CD στον 1D $CD[1000]$ και το πλήθος των τίτλων που κάθε κατάστημα κατέχει στον 2D $\Delta CD[1000,15]$
- Θα διαβάζει τον τίτλο ενός CD και θα εκτυπώνει τα καταστήματα στα οποία είναι διαθέσιμο
- Θα διαβάζει το όνομα ενός καταστήματος και θα εκτυπώνει το πλήθος των τίτλων που διαθέτει, εξαιρώντας τα μηδενικά

```
1 ΠΡΟΓΡΑΜΜΑ ΑΣΚΗΣΗ3
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: i, j, ΚΑΤCD[1000,15], pos, πλήθος
4   ΛΟΓΙΚΕΣ: done
5   ΧΑΡΑΚΤΗΡΕΣ: ΚΑΤ[15], CD[1000], ΤΙΤΛΟΣ, ΟΝΟΜΑΚΑΤ
6 ΑΡΧΗ
7   ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
8     ΓΡΑΨΕ 'Δώσε όνομα καταστήματος'
9     ΔΙΑΒΑΣΕ ΚΑΤ[i]
10  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
11
12  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 1000
13    ΓΡΑΨΕ 'Δώσε όνομα CD'
14    ΔΙΑΒΑΣΕ CD[i]
15    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 15
16      ΓΡΑΨΕ 'Πόσα αντίτυπα του δίσκου ', j
17      ΔΙΑΒΑΣΕ ΚΑΤCD[i,j]
18    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
19  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
20
```

```
21 ΓΡΑΨΕ 'Δώσε τίτλο CD'
22 ΔΙΑΒΑΣΕ ΤΙΤΛΟΣ
23 done <- ΨΕΥΔΗΣ
24 pos <- 0
25 i <- 1
26
27 ΟΣΟ done = ΨΕΥΔΗΣ ΚΑΙ i < 1000 ΕΠΑΝΑΛΑΒΕ
28   ΑΝ CD[i] = ΤΙΤΛΟΣ ΤΟΤΕ
29     done <- ΑΛΗΘΗΣ
30     pos <- i
31   ΑΛΛΙΩΣ
32     i <- i + 1
33   ΤΕΛΟΣ_ΑΝ
34 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
35
36 ΑΝ pos <> 0 ΤΟΤΕ
37   ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 15
38     ΑΝ ΚΑΤCD[pos, j] <> 0 ΤΟΤΕ
39       ΓΡΑΨΕ 'Είναι διαθέσιμο στο κατάστημα ', ΚΑΤ[i]
40     ΤΕΛΟΣ_ΑΝ
41   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
42 ΑΛΛΙΩΣ
43   ΓΡΑΨΕ 'Ο τίτλος δεν υπάρχει αυτή τη στιγμή σε κανένα δισκοπωλείο...'
44 ΤΕΛΟΣ_ΑΝ
45
```



```
46 ΓΡΑΨΕ 'Δώσε όνομα καταστήματος'
47 ΔΙΑΒΑΣΕ ΟΝΟΜΑΚΑΤ
48 pos <- 0
49 i <- 1
50 done <- ΨΕΥΔΗΣ
51 ΟΣΟ (i <= 15 ΚΑΙ done = ΨΕΥΔΗΣ) ΕΠΑΝΑΛΑΒΕ
52     ΑΝ ΚΑΤ[i] = ΟΝΟΜΑΚΑΤ ΤΟΤΕ
53         done <- ΑΛΗΘΗΣ
54         pos <- i
55     ΑΛΛΙΩΣ
56         i <- i + 1
57     ΤΕΛΟΣ_ΑΝ
58 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
59
60 ΑΝ pos <> 0 ΤΟΤΕ
61     πλήθος <- 0
62     ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 1000
63         ΑΝ ΚΑΤCD[i, pos] <> 0 ΤΟΤΕ
64             πλήθος <- πλήθος + 1
65     ΤΕΛΟΣ_ΑΝ
66     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
67     ΓΡΑΨΕ 'Το κατάστημα ', ΟΝΟΜΑΚΑΤ, ' παρέχει ', πλήθος, ' διαφορετικών τίτλων'
68 ΑΛΛΙΩΣ
69     ΓΡΑΨΕ 'Δεν υπάρχει αυτό το κατάστημα'
70 ΤΕΛΟΣ_ΑΝ
71
72 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
73
```

ΑΣΚΗΣΗ ΣΠΙΤΙ

- **ΑΝΤΙΜΕΤΑΘΕΣΗ ΠΙΝΑΚΩΝ – ΠΑΡΑΛΛΗΛΟΙ ΠΙΝΑΚΕΣ**
- **ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΚΑΤΑ ΓΡΑΜΜΕΣ ΚΑΙ ΣΤΗΛΕΣ ΣΕ ΔΙΣΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ**
- **ΕΥΡΕΣΗ ΠΛΗΘΟΥΣ**

Αγορά (σε €)	Άτοκες δόσεις
100 - 200	3
201 - 500	6
501 και πάνω	10

Τη στιγμή που η πωλήτρια του καταστήματος ρούχων περνά κάποιο προϊόν από το σαρωτή, η ταμειακή αναζητά στη ΒΔ τις πληροφορίες που αφορούν το προϊόν. Οι δομές είναι: 1D πίνακας Κ, με 2000 στοιχεία – όσα και τα προϊόντα – με τους κωδικούς τους, ένας παράλληλος πίνακας Τ με τις τιμές τους και παράλληλος πίνακας Π με τις σύντομες περιγραφές τους. ΝΑΠ που θα διαβάσει επαναληπτικά τον κωδικό του προϊόντος που περνά από το σαρωτή, θα εμφανίζει την περιγραφή του στην οθόνη και θα υπολογίζει το κόστος του στο συνολικό κόστος της αγοράς. Η διαδικασία θα ολοκληρώνεται όταν ως κωδικός εισαχθεί το #

Στο σημείο αυτό ο χρήστης θα ενημερώνεται ότι μπορεί να εξοφλήσει το ποσό σε άτοκες δόσεις, σύμφωνα με τον διπλανό πίνακα

Το πρόγραμμα θα πρέπει στο τέλος να εκτυπώνει το πλήθος των δόσεων, καθώς και το ποσό κάθε δόσης