

## ΑΣΚΗΣΕΙΣ ΣΤΟΙΒΑΣ

### ΑΣΚΗΣΗ 1

Στο παρακάτω πρόγραμμα, με τη χρήση των διαδικασιών ΩΘΗΣΗ και ΑΠΩΘΗΣΗ, υλοποιείται το εξής: ωθούνται 5 αριθμοί σε στοίβα και στη συνέχεια απωθούνται, και εμφανίζονται με τη σειρά που απωθούνται.

Να συμπληρώσετε τα κενά κατάλληλα ώστε να υλοποιούνται τα παραπάνω.

ΠΡΟΓΡΑΜΜΑ ΠΕΝΤΕ\_ΑΡΙΘΜΟΙ\_ΣΕ\_ΣΤΟΙΒΑ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: j, top, number, α

ΛΟΓΙΚΕΣ: done

Stack[100]

ΑΡΧΗ

top<-- 0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5

ΔΙΑΒΑΣΕ number

ΚΑΛΕΣΕ ΩΘΗΣΗ (Stack, \_\_\_\_(1)\_\_\_\_, top, done)

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5

ΚΑΛΕΣΕ ΑΠΩΘΗΣΗ (Stack, α, top, done)

ΓΡΑΨΕ \_\_\_\_(2)\_\_\_\_

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΩΘΗΣΗ (Stack, item, top, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Stack[100], item, top

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ top \_\_\_\_(3)\_\_\_\_ ΤΟΤΕ

top<-- \_\_\_\_(4)\_\_\_\_

\_\_\_\_(5)\_\_\_\_ <-- item

done<-- ΑΛΗΘΗΣ

ΑΛΛΙΩΣ

done<-- ΨΕΥΔΗΣ

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΑΠΩΘΗΣΗ (Stack, item, top, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Stack[100], item, top

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ top <> 0 ΤΟΤΕ

\_\_\_\_(6)\_\_\_\_ <-- Stack[top]

top<-- \_\_\_\_(7)\_\_\_\_

```
        done<-- ΑΛΗΘΗΣ
ΑΛΛΙΩΣ
        done<-- ΨΕΥΔΗΣ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

## ΑΣΚΗΣΗ 2

Δίνεται το παρακάτω κυρίως πρόγραμμα και δύο διαδικασίες. Αν για είσοδο δοθούν οι χαρακτήρες Α, Β, Χ, Δ, Α, Π, Μ, Ο, Π, Μ, Τ, Ζ να γράψετε στο τετράδιό σας τι θα εμφανιστεί στην οθόνη.

```
ΠΡΟΓΡΑΜΜΑ Β2
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: i, top
    ΛΟΓΙΚΕΣ: done
    ΧΑΡΑΚΤΗΡΕΣ: X, Stack[10]
ΑΡΧΗ
    top<-- 0
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 12
        ΔΙΑΒΑΣΕ X
        ΚΑΛΕΣΕ ΩΘΗΣΗ (Stack, X, top, done)
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 6
        ΚΑΛΕΣΕ ΑΠΩΘΗΣΗ (Stack, X , top, done)
        ΓΡΑΨΕ X
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
ΔΙΑΔΙΚΑΣΙΑ ΩΘΗΣΗ (Α, στοιχείο, top, done)
    ΣΤΑΘΕΡΕΣ
        N = 10
    ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: top
        ΧΑΡΑΚΤΗΡΕΣ: στοιχείο, Α[N]
        ΛΟΓΙΚΕΣ: done
    ΑΡΧΗ
        ΑΝ top < N ΤΟΤΕ
            top <-- top + 1
            Α[top] <-- στοιχείο
            done <-- ΑΛΗΘΗΣ
        ΑΛΛΙΩΣ
            done <-- ΨΕΥΔΗΣ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

ΔΙΑΔΙΚΑΣΙΑ ΑΠΩΘΗΣΗ (A, στοιχείο, top, done)

ΣΤΑΘΕΡΕΣ

N = 10

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: top

ΧΑΡΑΚΤΗΡΕΣ: στοιχείο, A[N]

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

AN top >= 1 TOTE

στοιχείο <-- A[top]

top <-- top - 1

done <-- ΑΛΗΘΗΣ

ΑΛΛΙΩΣ

done <-- ΨΕΥΔΗΣ

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ

### ΑΣΚΗΣΗ 3

Ένας εφοριακός υλοποιεί την εκκαθάριση φορολογικών δηλώσεων που καταφθάνουν στο γραφείο του, επεξεργαζόμενος κάθε φορά την τελευταία από αυτές που δέχτηκε.

Αυτή τη στιγμή έχει να επεξεργαστεί 40 δηλώσεις.

Στο γραφείο του μπορεί να υπάρχουν 100 δηλώσεις το πολύ.

Να γράψετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο θα υλοποιεί στοίβα με τη χρήση πίνακα ΔΗΛΩΣΕΙΣ[100] στο οποίο θα υλοποιούνται τα παρακάτω:

1. Θα διαβάζονται οι ΑΦΜ (αριθμοί φορολογικού μητρώου) των 40 δηλώσεων που έχει ήδη ο εφοριακός προς επεξεργασία, και θα τοποθετούνται στη στοίβα.
2. Θα διαβάζεται επαναληπτικά μια ενέργεια (Υ για παραλαβή νέας δήλωσης από τον εφοριακό και προσθήκη της στη στοίβα, Ε για εκκαθάριση υπάρχουσας δήλωσης της στοίβας και αφαίρεσή της από αυτήν) και θα εκτελεί την αντίστοιχη ενέργεια. Για κάθε εκκαθάριση να εμφανίζει το μήνυμα «ΕΚΚΑΘΑΡΙΣΗ ΔΗΛΩΣΗΣ» και τον αντίστοιχο ΑΦΜ.
3. Αν γίνουν 100 οι δηλώσεις, θα εμφανίζεται το μήνυμα «ΑΔΥΝΑΜΙΑ ΠΑΡΑΛΑΒΗΣ ΝΕΑΣ ΔΗΛΩΣΗΣ» και η επαναληπτική υποβολή/εκκαθάριση θα τερματίζει.
4. Αν ο εφοριακός ολοκληρώσει την εκκαθάριση όλων των δηλώσεων, θα εμφανίζεται το μήνυμα «ΕΠΙΤΥΧΗΣ ΟΛΟΚΛΗΡΩΣΗ ΕΚΚΑΘΑΡΙΣΗΣ» και η επαναληπτική υποβολή/εκκαθάριση θα τερματίζει.
5. Θα εμφανίζεται το πλήθος των δηλώσεων που εκκαθαρίστηκαν.

#### ΑΣΚΗΣΗ 4

Ένα εργοστάσιο υλοποιεί την παραγωγή προϊόντων του σε 2 στάδια (Α,Β). Το προϊόν αρχικά συναρμολογείται στο στάδιο Α, και στη συνέχεια προωθείται και συσκευάζεται στο στάδιο Β. Η λογική που ακολουθείται στα 2 στάδια παραγωγής είναι η λογική της στοίβας (το τελευταίο προϊόν που εισέρχεται, είναι το πρώτο που εξέρχεται - LIFO). Τα 2 στάδια παραγωγής μπορούν να περιέχουν το πολύ 200 προϊόντα το καθένα.

Όμως, για λόγους ομαλής ροής της παραγωγής, δε μπορεί σε μια από τις 2 στοίβες να υπάρχουν πάνω από τα διπλάσια προϊόντα σε σχέση με την άλλη.

Να γράψετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο θα υλοποιεί τη γραμμή παραγωγής ως εξής:

Α) Να διαβάσει το πλήθος των στοιχείων που βρίσκονται αυτή τη στιγμή σε καθεμιά από τις 2 στοίβες, ελέγχοντας ότι καμία από τις 2 στοίβες δεν έχει πάνω από το διπλάσιο αριθμό προϊόντων σε σχέση με την άλλη, και ότι καμία από τις 2 στοίβες δεν είναι αρχικά γεμάτη ή άδεια.

Β) Να διαβάσει τον κωδικό προϊόντος των στοιχείων για καθένα από τα αρχικά στοιχεία της στοίβας Α[200], και να τα τοποθετεί στη στοίβα. Ομοίως για τη Β[200].

Γ) Να διαβάσει επαναληπτικά το είδος της ενέργειας που πρόκειται να γίνει ως εξής:

«Α» για ώθηση στη στοίβα Α (είσοδος στη γραμμή παραγωγής)

«ΑΒ» για απώθηση από τη στοίβα Α και ώθηση στη στοίβα Β (συναρμολόγηση)

«Β» για απώθηση από τη στοίβα Β (συσκευασία)

Η διαδικασία τερματίζει όταν συμβεί αποτυχία εκτέλεσης ενέργειας 5 συνεχόμενες φορές (ερώτημα Δ)

Δ) Να εκτελεί τις παραπάνω ενέργειες (ωθήσεις – απωθήσεις), και να εμφανίζει το μήνυμα «ΑΠΟΤΥΧΙΑ ΕΚΤΕΛΕΣΗΣ ΕΝΕΡΓΕΙΑΣ» κάθε φορά που έχουμε υπερχειλίση, υποχειλίση κάποιας στοίβας ή παραβιάζεται ο κανόνας ομαλής ροής της παραγωγής (δε μπορεί μια στοίβα να έχει υπερδιπλάσια προϊόντα από την άλλη).

Διαφορετικά να εμφανίζει αντίστοιχα μηνύματα:

«ΕΠΙΤΥΧΗΣ ΕΙΣΟΔΟΣ»

«ΕΠΙΤΥΧΗΣ ΣΥΝΑΡΜΟΛΟΓΗΣΗ»

«ΕΠΙΤΥΧΗΣ ΣΥΣΚΕΥΑΣΙΑ»

συνοδευόμενα από τον κωδικό προϊόντος, για καθεμιά από τις ενέργειες Α, ΑΒ, Β αντίστοιχα.

Προφανώς η ενέργεια ΑΒ συμβαίνει μόνο αν είναι δυνατές και οι 2 ενέργειες (απόθεση Α – ώθηση Β), αλλιώς δε συμβαίνει καθόλου.

Ε) Το πρόγραμμα να υπολογίζει και να εμφανίζει:

1. Το πλήθος των προϊόντων που εισήλθαν στη γραμμή παραγωγής
2. Το πλήθος των προϊόντων που παρήχθησαν
3. Το ποσοστό των αποτυχημένων ενεργειών
4. Το πλήθος των φορών που συνέβησαν 3 συνεχόμενες αποτυχημένες εκτελέσεις ενέργειας