

1.

Έλεγχος ροής

1.1. Δομή επιλογής

Συνήθως τα προβλήματα έχουν κάποιες ιδιαιτερότητες και δεν ισχύουν τα ίδια βήματα επίλυσης με βήματα στη σειρά, αλλά χρειάζεται να λαμβάνονται κάποιες αποφάσεις με βάση κάποια δεδομένα κριτήρια, που μπορεί να είναι διαφορετικά σε κάθε βήμα του προβλήματος.

Γενικά η διαδικασία της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ/ ΙΣΧΥΕΙ ή ΔΕΝ ΙΣΧΥΕΙ) και ακολουθεί η απόφαση εκτέλεσης κάποιας ενέργειας με βάση την τιμή της λογικής αυτής συνθήκης.

Οι δομές επιλογής διακρίνονται σε:

- Δομή απλής επιλογής
- Δομή σύνθετης επιλογής
- Δομή πολλαπλής επιλογής
- Δομή εμφωλευμένης επιλογής

1.2. Δομή απλής επιλογής

Γενική μορφή απλής επιλογής

if (συνθήκη) εντολή

Λειτουργεί ελέγχοντας εάν ισχύει η συνθήκη, δηλαδή είναι αληθής, τότε μόνο εκτελείται η εντολή. Σε κάθε άλλη περίπτωση εκτελείται η εντολή που ακολουθεί μετά το τέλος της.

Εάν εκτελεί μια μόνο εντολή δεν κλείνει με την εντολή αγκύλη {}, ενώ εάν εκτελεί περισσότερες από μια εντολές, όπως στη συνέχεια χρειάζεται συντάσσεται:

```
if (συνθήκη) {  
    Εντολές  
}
```

1.3. Δομή Σύνθετης Επιλογής

Συχνά η εντολή if εκτός από το τμήμα των εντολών, που εκτελούνται όταν η λογική έκφραση είναι αληθής, περιέχει και το τμήμα των εντολών που εκτελούνται, αν δεν ισχύει η συνθήκη, δηλαδή είναι ψευδής. Έχει τη μορφή:

```
if (συνθήκη)  
    εντολή  
else  
    εντολή
```

για μια εντολή ανά τμήμα και για περισσότερες:

```
if (συνθήκη) {  
    Σύνολο_εντολών_1  
} else {  
    Σύνολο_εντολών_2  
}
```

Αν η συνθήκη ισχύει, δηλαδή είναι αληθής, τότε εκτελούνται οι εντολές που βρίσκονται μεταξύ των πρώτων {}, διαφορετικά εκτελούνται οι εντολές μεταξύ των {} του else. Η εκτέλεση συνεχίζεται με την εντολή που ακολουθεί την τελευταία }.

Στη σύνθετη επιλογή εκτελούνται εντολές στο αληθής και στο ψευδής της συνθήκης, ενώ στην απλή μόνο στο αληθής. Τόσο η απλή επιλογή όσο και η σύνθετη ελέγχουν μία μόνο συνθήκη και εκτελείται είτε το τμήμα του αλγορίθμου που βρίσκεται μέσα στις πρώτες {} ή μεταξύ των δεύτερων {}, ΔΗΛΑΔΗ ένα από τα δύο (ποτέ και τα δύο) και στη συνέχεια ο έλεγχος του αλγορίθμου περνάει αμέσως μετά το τέλος της αλγορίθμικής δομής της Σύνθετης Επιλογής για την εκτέλεση των επόμενων εντολών του προγράμματος.

1.4. Δομή Πολλαπλής Επιλογής

Οι διαδικασίες των πολλαπλών επιλογών εφαρμόζονται στα προβλήματα όπου μπορεί να ληφθούν διαφορετικές αποφάσεις ανάλογα με την τιμή που μπορεί να παίρνει μία έκφραση. Η έκφραση μπορεί να πάρει πάνω από δύο διαφορετικές τιμές.

```
if (συνθήκη1)
    εντολή_1
else if (συνθήκη_2)
    εντολή_2
else if (συνθήκη3)
    εντολή_3
...
else
    εντολή_ν
```

για μια εντολή ανά τμήμα και για περισσότερες:

```
if (συνθήκη1) {
    Σύνολο_εντολών_1
} else if (συνθήκη_2) {
    Σύνολο_εντολών_2
} else if (συνθήκη3) {
    Σύνολο_εντολών_3
}
...
} else {
    Σύνολο_εντολών_ν
}
```

Εκτελούνται οι εντολές που βρίσκονται στο αντίστοιχο τμήμα, όταν η συνθήκη είναι αληθής. Η εκτέλεση του προγράμματος συνεχίζεται με την τελευταία αγκύλη. Εκτελείται ΠΑΝΤΟΤΕ ΜΟΝΟ μία ομάδα εντολών. Επομένως Όλες οι προηγούμενες συνθήκες- από αυτήν που εκτελείται- ΠΡΕΠΕΙ ΝΑ είναι ψευδείς. Επιπρόσθετα, οφείλουμε πάντοτε να προσέχουμε τη διάταξη των <Συνθηκών> έτσι ώστε να μην αποκλείσουμε- από λογικό λάθος- κάποια.

1.5. Δομή Πολλαπλής Επιλογής switch

Αν για την ίδια μεταβλητή ή έκφραση είναι αναγκαίο να ελεγχθούν πολλές διαφορετικές τιμές, επιλογές ή περιπτώσεις, τότε μπορεί (ή είναι προτιμότερο) να χρησιμοποιηθεί το σχήμα πολλαπλής επιλογής switch, αντί για if.

Στην πολλαπλή επιλογή switch, ανάλογα με την τιμή της μεταβλητής ή έκφρασης εκτελούνται οι εντολές μετά την case που αντιστοιχεί στην τιμή της μεταβλητής ή της έκφρασης. Οι τιμές που συνοδεύουν κάθε case μπορεί να είναι μία ή περισσότερες διακριτές τιμές, περιοχή τιμών ή να υπακούουν σε μια συνθήκη. Αν η τιμή της

μεταβλητής ή της έκφρασης δεν αντιστοιχεί σε καμία case, τότε εκτελούνται οι εντολές που ακολουθούν την default. Μετά την εκτέλεση μιας από τις περίπτωσης (και μόνο μιας από όλες), ο αλγόριθμος συνεχίζει με την εντολή, που ακολουθεί την τελευταία αγκύλη.

```
switch (μεταβλητή ή έκφραση) {
    case <τιμή_1>:
        Εντολές_1;
        break;
    case <τιμή_2>:
        Εντολές_2;
        break;
    ...
    default:
        Εντολές_v;
        break;
}
```

1.6. Εμφωλευμένη Δομή Επιλογής

Όταν οι εναλλακτικές περιπτώσεις είναι περισσότερες από δύο, τότε μπορούν να χρησιμοποιηθούν πολλές επιλογές, η μία μέσα στην άλλη, που ονομάζονται εμφωλευμένες. Επιβάλλεται εξαιρετικά μεγάλη προσοχή στην ολοκλήρωση των δομών, ώστε να αποφεύγεται η επικάλυψη των εμφωλευμένων δομών επιλογή. Δηλαδή θα πρέπει να ολοκληρώνεται πρώτη η επιλογή που ξεκίνησε τελευταία. Η σύνταξη είναι όπως η ακόλουθη:

```
if (συνθήκη_1) {
    if (συνθήκη_2) {
        .....
    }
    .....
}
```

1.7. Ασκήσεις

- Γράψτε ένα πρόγραμμα, όπου θα εισάγετε τις πλευρές ενός τριγώνου και θα ελέγχετε εάν το τρίγωνο είναι έγκυρο ή όχι.

Υπόδειγμα: Ένα τρίγωνο είναι έγκυρο, αν το άθροισμα των δύο πλευρών του είναι μεγαλύτερο από την τρίτη πλευρά. Έστω ότι τα a, b, c είναι τρεις πλευρές ενός τριγώνου. Τότε το τρίγωνο είναι έγκυρο, αν πληρούνται και οι τρεις προϋποθέσεις: $a + b > c$, $a + c > b$ και $b + c > a$

2. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει την ώρα σε μορφή hh:mm:ss και να εμφανίζει πόση ώρα απομένει μέχρι τα μεσάνυχτα (δηλ. 24:00:00).
3. Ένα εργοστάσιο παρασκευάζει μικρά και μεγάλα μπουκάλια εμφιάλωσης. Τα μικρά κοστίζουν 0.008€ και τα μεγάλα 0.02€. Για παραγγελίες μεγαλύτερες από 200€ ή 3000 τεμάχια, υπάρχει έκπτωση 8%. Για παραγγελίες μεγαλύτερες από 600€, η έκπτωση είναι 20%. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει το πλήθος των μικρών και μεγάλων μπουκαλιών μίας παραγγελίας και να εμφανίζει το συνολικό κόστος.
4. Να γραφεί πρόγραμμα που δέχεται ένα πενταψήφιο θετικό ακέραιο, αναστρέφει τα ψηφία του και τυπώνει το αποτέλεσμα. Χρησιμοποιήστε την πρόταση while και τον τελεστή %, και κάνετε αμυντικό προγραμματισμό για να ελέγχετε ότι ο ακέραιος είναι θετικός και πενταψήφιος.
5. Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει τον αριθμό ενός μήνα (Ιαν=1,..., Δεκ=12), τον αριθμό της ημέρας (π.χ. αν ο μήνας είναι ο Ιανουάριος οι έγκυρες τιμές είναι 1...31) και να εμφανίζει την ημερομηνία μετά από 50 ημέρες. Στην περίπτωση του Φεβρουαρίου, το πρόγραμμα να προτρέπει τον χρήστη να εισάγει τις ημέρες του, δηλαδή 28 ή 29. Για παράδειγμα, αν ο χρήστης εισάγει για μήνα την τιμή 3 και αριθμός ημέρας 5, το πρόγραμμα να εμφανίζει 4/24.
6. Κάθε Ethernet κάρτα δικτύου χαρακτηρίζεται από έναν παγκόσμια μοναδικό 48-bit αριθμό που ονομάζεται MAC διεύθυνση. Συνήθως, η MAC διεύθυνση γράφεται σε δεκαεξαδική μορφή. Για να βρούμε τον τύπο της διεύθυνσης ελέγχουμε την τιμή της πρώτης οκτάδας από αριστερά. Αν η τιμή της είναι άρτια, η διεύθυνση είναι σημείο-προς-σημείο (unicast). Αν είναι περιττή, η διεύθυνση είναι σημείο-προς-πολλά σημείο (multicast). Αν όλες οι οκτάδες είναι ίσες με 0xFF, η διεύθυνση είναι ευρείας εκπομπής (broadcast). Για παράδειγμα:
 - α) Η διεύθυνση FF:FF:FF:FF:FF:FF είναι broadcast.
 - β) Η διεύθυνση 18:20:3F:20:AB:11 είναι unicast, γιατί το 18 είναι άρτιος αριθμός.
 - γ) Η διεύθυνση A3:3F:40:A2:C3:42 είναι multicast, γιατί το A3 είναι περιττός αριθμός.

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει μία MAC διεύθυνση και να εμφανίζει τον τύπο της. Θεωρήστε ότι ο χρήστης θα εισάγει την MAC όπως και στα προηγούμενα παραδείγματα, δηλαδή με τη μορφή x:x:x:x:x, όπου το κάθε x είναι ένας δεκαεξαδικός αριθμός.