Πανεπιστήμιο Αιγαίου

Εισαγωγή στην Πληροφορική

Ενότητα 3: Προγραμματισμός ηλεκτρονικών υπολογιστών

Μιχάλης Βαΐτης
Τμήμα Γεωγραφίας

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σκοποί ενότητας

- Να γνωρίζετε την έννοια του αλγορίθμου.
- Να μπορείτε να χρησιμοποιείτε τις αλγοριθμικές δομές της διαδοχικότητας, της απόφασης και της επανάληψης για την επίλυση απλών υπολογιστικών προβλημάτων.
- Να μπορείτε να καταγράφετε έναν αλγόριθμό με λογικό διάγραμμα.
- Να γνωρίζετε τι είναι γλώσσα προγραμματισμού και τη διαδικασία κατασκευής ενός προγράμματος.
- Να μπορείτε να κατασκευάζεται προγράμματα για την επίλυυση απλών υπολογιστικών προβλημάτων με τη γλώσσα προγραμματισμού R.

Υπολογισμός δύναμης

Η ύψωση σε δύναμη είναι μαθηματική πράξη, που συμβολίζεται ως α^n και περιλαμβάνει δύο αριθμούς, την βάση α και τον εκθέτη n.

Αν το n είναι θετικός ακέραιος, η ύψωση σε δύναμη αντιστοιχεί σε επαναλαμβανόμενο πολλαπλασιασμό, με άλλα λόγια είναι το γινόμενο n παραγόντων α:

$$a^n = \underbrace{a \times \cdots \times a}_{n},$$

Αν το n είναι μηδέν, το αποτέλεσμα είναι ο αριθμός 1. Δηλ. $\alpha^0 = 1$

Υπολογισμός δύναμης

Αν α και n είναι θετικοί ακέραιοι ή μηδέν, γράψτε πρόγραμμα στην R που να υπολογίζει το α^n, χωρίς τη χρήση του τελεστή ^.

```
print('Δώσε βάση')
a <- as.numeric(readline())
print('Δώσε εκθέτη')
n <- as.numeric(readline())
power <- 1
if (n == 0) {
     print(power)
} else {
     for (i in 1:n) power <- power * a
     print(power)
     }
```

Υπολογισμός δύναμης

Εναλλακτικά, με μόνο μία εντολή print για το αποτέλεσμα. Δεν εκτελούνται εντολές στο if αν η συνθήκη είναι αληθής. Στο else υπάρχει μόνο μία εντολή – το for, το οποίο με τη σειρά του εκτελεί επαναληπτικά μόνο ένα υπολογισμό.

```
print('Δώσε βαση')
a <- as.numeric(readline())
print('Δώσε εκθέτη')
n <- as.numeric(readline())
power <- 1
if (n == 0) {
} else for (i in 1:n) power <- power * a
print(power)
```

Επίδοση αθλητή

Η επίδοση ενός αθλητή υπολογίζεται ως ο μέσος όρος της βαθμολογίας των έγκυρων προσπαθειών του. Αν η προσπάθεια είναι έγκυρη, βαθμολογείται με έναν αριθμό στο διάστημα [1, 10]. Αν η προσπάθεια είναι άκυρη, βαθμολογείται με μηδέν. Ο αθλητής κάνει πάντα τρεις προσπάθειες.

Π.χ.

Αν κάνει τρεις προσπάθειες με βαθμολογία 0, 5, 7 τότε η επίδοση είναι 6 (δηλ. (5+7)/2).

Αν κάνει τρεις προσπάθειες με βαθμολογία 5, 8, 8 τότε η επίδοση είναι 7 (δηλ. (5+8+8)/3).

Επίδοση αθλητή

Γράψτε πρόγραμμα που να υπολογίζει την επίδοση ενός αθλητή για τρεις προσπάθειες.

```
a <- as.numeric(readline())
b <- as.numeric(readline())
c <- as.numeric(readline())
n <- 3
if (a == 0) n <- n-1
if (b == 0) n <- n-1
if (c == 0) n <- n-1
if (n ==0) {
      print(0)
} else {
      print((a+b+c)/n)
}
```

Υποπρογράμματα

Υποπρογράμματα (ή Διαδικασίες):
Συναρτήσεις και Υπορουτίνες

Αποτελούν τμήματα κώδικα που δέχονται προκαθορισμένα δεδομένα εισόδου, εκτελούν μια συγκεκριμένη εργασία και (ενδεχομένως) παράγουν δεδομένα εξόδου. Στη συνέχεια μπορούν να *κληθούν* από άλλα σημεία του προγράμματος.

Εισάγουν την έννοια της αφαίρεσης (abstraction) στον προγραμματισμό (έμφαση στο *τι* και όχι στο *πως*)

<u>Πλεονεκτήματα:</u>

- Δομημένος προγραμματισμός
- Επαναχρησιμοποίηση κώδικα
- Ευκολότερη διόρθωση του κώδικα
- Συνεργασία προγραμματιστών
- Επέκταση της γλώσσας

Υποπρογράμματα στην R

```
όνομα = Function ( ορίσματα ) {
        εντολές
        return( τιμή )
}
```

Τα ορίσματα αποτελούν τις μεταβλητές εισόδου της συνάρτησης. Η ανάθεση τιμής γίνεται κατά την κλήση της συνάρτησης.

Αν υπάρχει η εντολή return, τότε το υποπρόγραμμα επιστέφει την τιμή στη θέση του κώδικα απ' όπου καλείται (δηλαδή λειτουργεί ως συνάρτηση/function).

Αν δεν υπάρχει η εντολή return, τότε το υποπρόγραμμα επιστέφει την τελευταία έκφραση που έχει υπολογιστεί (δηλαδή λειτουργεί ως συνάρτηση/function).

Αλλιώς , το υποπρόγραμμα επιστέφει την κενή τιμή NULL (δηλαδή λειτουργεί ως υποροutίνα/subroutine).

Παράδειγμα συνάρτησης

```
decimal_degrees = function (m, l, d) {
  if (m>=0 && m<=359 && l>=0 && l<=59 && d>=0
&& d<=59) dd=m + l/60 + d/3600 else dd=-1
    return(dd)
}

a <- as.numeric(readline())
b <- as.numeric(readline())
c <- as.numeric(readline())
print(decimal_degrees(a, b, c))
```

Τέλος Ενότητας