

### ΤΥΧΑΙΑ ΔΕΔΟΜΕΝΑ

Βασική πράξη αποτελεί η δημιουργία τυχαίων αριθμών. Η δημιουργία τυχαίων αριθμών επιτυγχάνεται με δυο τρόπους:

- I) με μαθηματικές συναρτήσεις που παράγουν τους λεγόμενους "ψευδοτυχαίους" αριθμούς και
- II) με πειράματα τύχης: ρίψη ενός νομίσματος, χρήση των ενδείξεων του εσωτερικού ρολογιού ενός ΗΥ κλπ.

#### I) Μαθηματικές συναρτήσεις:

α) Στις περισσότερες γλώσσες προγραμματισμού υπάρχουν εντολές (ουσιαστικά έτοιμες συναρτήσεις) οι οποίες επιστρέφουν ψευδοτυχαίες τιμές σε ένα ορισμένο διάστημα. π.χ. στην BASIC η εντολή  $X=RND$  εκχωρεί στη μνήμη  $X$  έναν αριθμό  $a$  με  $0 < a < 1$ . Στην PASCAL η εντολή  $X:=Random$  εκχωρεί στη μνήμη  $X$  έναν δεκαδικό αριθμό  $a$  με  $0 < a < 1$  και η εντολή  $X:=Random(1000)$  εκχωρεί στη μνήμη  $X$  έναν ακέραιο αριθμό  $a$  με  $0 < a < 1000$ . Όμως οι αριθμοί αυτοί είναι ψευδοτυχαίοι, που σημαίνει ότι στην πραγματικότητα παράγονται από μια συνάρτηση και μετά από έναν ορισμένο αριθμό επαναλαμβάνονται (δηλαδή η συνάρτηση παραγωγής τους είναι περιοδική). Επιπλέον σε κάθε εκτέλεση του προγράμματος παράγονται οι ίδιοι τυχαίοι αριθμοί. Για παράδειγμα η εντολή:

```
for i:=1 to 10 do writeln(Random(1000))
```

παράγει πάντοτε τους αριθμούς 0, 31, 861, 202, 272, 671, 318, 161, 372, 425. Παρ' όλα αυτά οι ψευδοτυχαίοι αριθμοί είναι χρήσιμοι αφού ικανοποιούν αρκετά στατιστικά tests τυχαιότητας.

Σημείωση: Την ιδιότητα των ψευδοτυχαίων να δίνουν πάντοτε τα ίδια αποτελέσματα για τις ίδιες παραμέτρους, την εκμεταλλευόμαστε ιδιαίτερα στην κατασκευή πολύπλοκων προγραμμάτων αφού η ιδιότητα αυτή ισοδυναμεί με την ύπαρξη δεδομένων και σταθερών αρχικών τιμών.

β) Συναρτήσεις που κατασκευάζονται από το χρήστη<sup>1</sup>:

Η προσφιλέστερη μέθοδος έγκειται στην επιλογή τεσσάρων αριθμών:  $a, c, m, x_0$ . Η παραγωγή τυχαίων αριθμών γίνεται με τη βοήθεια της αναδρομικής σχέσης:  $x_{n+1} = (ax_n + c) \bmod m$ . Η επιλογή των αριθμών αυτών είναι αρκετά πολύπλοκη υπόθεση. Πρακτικά ισχύουν οι ακόλουθοι κανόνες:

- 1) ο  $m$  πρέπει να είναι "μεγάλος" και μάλιστα κατά προτίμηση δύναμη του 10 ή του 2
- 2) ο  $c$  μπορεί να είναι 1
- 3) ο  $b$  πρέπει να είναι της μορφής  $k^{21}$  όπου  $k$  είναι ένας άρτιος αριθμός κι επιπλέον καλό είναι να έχει ένα ψηφίο λιγότερο από τον  $m$
- 4) η επιλογή των συντελεστών πρέπει επίσης να λαμβάνει υπόψη της την υπερχειλίση. Για να αποφευχθούν οι ενδιάμεσες υπερχειλίσεις χρησιμοποιείται το εξής τέχνασμα<sup>2</sup>:

Για την εκτέλεση του πολλαπλασιασμού  $p * q$ , θεωρούμε ότι

$$p = 10^4 * p_1 + p_0 \text{ και } q = 10^4 * q_1 + q_0$$

$$\text{Άρα } p * q = (10^4 * p_1 + p_0) * (10^4 * q_1 + q_0) = 10^8 * p_1 * q_1 + 10^4 * (p_1 * q_0 + p_0 * q_1) + p_0 * q_0$$

---

<sup>1</sup> Για μια ανάλυση των σχετικών αλγορίθμων δεξ D.E. Knuth, The Art Of Computer Programming, Vol II, Seminumerical Algorithms, Addison-Wesley, που αποτελεί βασική αναφορά στη διεθνή βιβλιογραφία.

<sup>2</sup> R. Sedgewick, Algorithms, Addison-Wesley, σ. 533

Ένα παράδειγμα εφαρμογής αποτελεί το παρακάτω πρόγραμμα:

```
program randomnumbers;  
const  
    m=100000000;  
    m1=10000;  
    b=31415821;  
var  
    i, a, N: integer;  
  
procedure mult(p, q: integer; var multres: integer);  
var  
    p0, q0, p1, q1: integer;  
begin  
    p1 := p div m1; p0 :=p mod m1;  
    q1 := q div m1; q0 :=q mod m1;  
    multres:=((p0*q1+p1*q0) mod m1)*m1 + p0*q0) mod m;  
end;  
  
procedure random(var x: integer);  
var  
    multres: integer;  
begin  
    mult(a,b, multres);  
    x:=(multres+1) mod m;  
end;  
  
Begin  
    readln(N); readln(a)  
    for i:=1 to N do begin  
        random(a);  
        writeln(a);  
    end;  
End.
```

## II) Πειράματα τύχης:

- α)** Οι περισσότερες γλώσσες προγραμματισμού παρέχουν τη δυνατότητα παραγωγής πραγματικών τυχαίων αριθμών. Στην PASCAL χρησιμοποιείται η διαδικασία Randomize η οποία σε συνδυασμό με την Random παράγει πραγματικούς τυχαίους αριθμούς.
- β)** Αν οι αριθμοί που παράγονται είναι στο διάστημα  $[0, 1)$  -που αποτελεί την πιο συνηθισμένη περίπτωση- κι επιθυμούμε να παράγουμε τυχαίους αριθμούς στο διάστημα  $[A, B]$ , όπου  $A, B$  ακέραιοι και  $A < B$ , τότε χρησιμοποιούμε τον τύπο:

$$Y = \text{trunc}((B-A)*L + A + 0.5)$$

όπου  $L$  ο τυχαίος αριθμός που παρήγαγε το σύστημα (από το βιβλίο του Κ. Κ. Τσούρου, σελ.26).

**ΑΣΚΗΣΕΙΣ.** Να γραφούν προγράμματα για κάθε ένα από τα παρακάτω προβλήματα:

1. Να παραχθούν και να εμφανιστούν 100 τυχαίοι ακέραιοι αριθμοί στο διάστημα  $[0, 1000)$ .
2. Να παραχθούν και να εμφανιστούν 100 ακέραιοι τυχαίοι στο διάστημα  $[1, 1000)$ .
3. Να παραχθούν και να εμφανιστούν 100 τυχαίες λέξεις που να αποτελούνται από κεφαλαία αγγλικά γράμματα και να έχουν μήκος 10 χαρακτήρων.
4. Να παραχθούν και να εμφανιστούν 100 δεκαδικοί τυχαίοι στο διάστημα  $[0, 1000)$ . Κάθε αριθμός θα χρησιμοποιεί στοιχίση των 5 χαρακτήρων με ακρίβεια τριών δεκαδικών ψηφίων. Δηλαδή αριθμός:5:3, οπότε οι αριθμοί θα εμφανίζονται στοιχημένοι στα αριστερά χωρίς κενά.

5. Να παραχθούν και να εμφανιστούν 100 τυχαία ονοματεπώνυμα (ουσιαστικά τυχαίες λέξεις κεφαλαίων αγγλικών χαρακτήρων με πρώτο το επώνυμο) με συνολικό μήκος από 15 ως 25 χαρακτήρες στα οποία το επίθετο να είναι το πολύ 12 χαρακτήρων. Η εμφάνιση κάθε ονοματεπώνυμου να γίνεται σε ξεχωριστή γραμμή και να σημειώνεται πρώτα το επίθετο και μετά το όνομα. Στους 25 χαρακτήρες περιλαμβάνεται και ο κενός ' ' μεταξύ επωνύμου και ονόματος.
6. Να κατασκευαστεί και να εμφανιστεί ένας πίνακας 100 θέσεων με τυχαίους ακεραίους εκ των οποίων ο πρώτος να είναι στο διάστημα [0,1000). Τα στοιχεία του πίνακα θα πρέπει να είναι ταξινομημένα σε αύξουσα σειρά.
7. Να παραχθούν και να εμφανιστούν 100 ακέραιοι τυχαίοι στο διάστημα [0, 1000) χωρίς επαναλήψεις. Χρησιμοποιείστε τη μέθοδο **β** όπως αυτή περιγράφεται παρακάτω.  
Σημείωση: Το πρόβλημα της παραγωγής χωρίς επαναλήψεις απαιτεί μια ιδιαίτερη αντιμετώπιση όπως φαίνεται παρακάτω:
- α) Η μέθοδος με έλεγχο μπορεί να είναι εξαιρετικά χρονοβόρα. Είναι ακόμη δυνατό και να πέσει σε infinite loop αν χρησιμοποιηθεί ένας αλγόριθμος παραγωγής ψευδοτυχαίων όχι καλής ποιότητας.
- β) Αν το εύρος των τυχαίων αριθμών είναι σχετικά μικρό ( $N$ =εύρος των τυχαίων αριθμών,  $M$ =πλήθος τυχαίων αριθμών), τότε το πρόβλημα μπορεί να αντιμετωπιστεί με τον εξής τρόπο:

```
for i:=1 to N do Pin[i]:=i-1;
for i:= 1 to N do begin
  A:=Pin[random(N) +1];
  Pin[random(N) +1]:=Pin[N];
  N:=N-1;
end;
```

- γ) Αν το εύρος των τυχαίων αριθμών είναι σχετικά μεγάλο τότε μπορεί να χρησιμοποιηθεί ο εξής αλγόριθμος<sup>3</sup>:

```
Θέσε S το κενό σύνολο
for j:=N-M+1 to N do begin
  T:= ένας τυχαίος ακέραιος στο διάστημα [1, j];
  if T not in S then insert T in S
  else insert j in S;
end;
```

Ενδιαφέρον σημείο είναι ο έλεγχος του αν το T ανήκει στο σύνολο S ή όχι. Κατάλληλες δομές δεδομένων είναι οι πίνακες Boolean (αν το N δεν είναι υπερβολικά μεγάλο) ή οι συνδεδεμένες λίστες.

---

<sup>3</sup> Jon Bentley, More Programming Pearls, σελ. 139-144