

Παραδείγματα Εισόδου - Εξόδου- Καθοριστικότητας – Αποτελεσματικότητας - Περατότητας

Είσοδος

1. Κάθε αλγόριθμος απαραίτητα ικανοποιεί τα επόμενα κριτήρια (ακολουθεί πρώτη η είσοδος)
Συνεπώς ΚΑΘΕ αλγόριθμος ΠΡΕΠΕΙ να έχει ΕΙΣΟΔΟ

Η πρόταση «Η είσοδος σε έναν αλγόριθμο είναι πάντα απαραίτητη» είναι ΣΩΣΤΗ και δεν υπάρχει διφορούμενο σημείο.

Στο βιβλίο ακολουθεί ο ορισμός της εισόδου:

2. Καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο.
Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών

Στην περίπτωση που δεν δίνεται καμία τιμή δεδομένων (από τον χρήστη) τότε παράγεται κάποια. Αυτό που έχει σημασία είναι πως ΠΑΝΤΑ υπάρχει είσοδος.

Ο παρακάτω αλγόριθμος έχει είσοδο;

Αλγόριθμος Άσκηση

$X \leftarrow 2$

$Y \leftarrow 2 * X$

Εμφάνισε Y

Τέλος Άσκηση

Έξοδος

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10

i <- i + 1

ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ

ΠΡΕΠΕΙ απαραίτητα να ικανοποιείται το κριτήριο της εξόδου και προφανώς στη συγκεκριμένη περίπτωση η έξοδος είναι ο μετρητής.

Επομένως σε πρόγραμμα πρέπει να χρησιμοποιηθεί Η ΜΟΝΗ εντολή εξόδου (ΓΡΑΨΕ).

Στην ψευδογλωσσα όμως θα χρησιμοποιηθεί η εντολή Εκτύπωσε, "Αποτελέσματα", που ακριβώς αυτός είναι ο ρόλος της.

Καθοριστικότητα

Καθοριστικότητα είναι το να είναι τα πάντα στον αλγόριθμο καθορισμένα και να ξέρει κάθε στιγμή αυτός που εκτελεί τον αλγόριθμο τι πρέπει να κάνει. Δηλαδή να μην έχει απορίες της μορφής «τι πρέπει να κάνω τώρα;» Υπάρχουν 2 περιπτώσεις παραβίασης της καθοριστικότητας:

Περίπτωση 1: Ο αλγόριθμος να μην περιέχει όλες τις δυνατές περιπτώσεις που μπορεί να προκύψουν. Παράδειγμα είναι η διαίρεση (όπως λέει και το βιβλίο). Δηλαδή να μην εξετάζεται η περίπτωση ο διαιρέτης να είναι μηδέν. Τέτοιου είδους παραβίαση της καθοριστικότητας οδηγεί σε σφάλμα κατά την εκτέλεση του αλγορίθμου (αν προκύψει η συγκεκριμένη περίπτωση).

Περίπτωση 2: Τα βήματα που περιέχονται να μην είναι σαφώς καθορισμένα. Παράδειγμα είναι σε μια μαγειρική συνταγή να περιέχεται η εντολή «προσθέστε λίγο αλάτι».

Τι πάει να πει λίγο; Το λίγο είναι σχετικό. Ένας υπολογιστής δεν καταλαβαίνει τι σημαίνει λίγο. Δηλαδή έχουμε ασάφεια. Τέτοιου είδους παραβίαση οδηγεί σε μη εκτελέσιμο αλγόριθμο. Στη γενική περίπτωση αυτός που εκτελεί τον αλγόριθμο δε θα ξέρει τι πρέπει να κάνει εκείνη τη στιγμή. Συγκεκριμένα στον υπολογιστή, όταν έρθει η ώρα ο αλγόριθμος να γίνει πρόγραμμα σε κάποια γλώσσα δε θα υπάρχει κάποια αντίστοιχη εντολή γλώσσας που να μπορεί να γίνει κατανοητή από τον υπολογιστή. Αυτή η περίπτωση λύνεται αν χρησιμοποιούνται οι εντολές που περιέχει το βιβλίο. Οι εντολές αυτές μόνο τυχαίες δεν είναι. Είναι αυτές που υποστηρίζονται από τις πραγματικές γλώσσες προγραμματισμού και έτσι η μετάβαση από τον αλγόριθμο στην πραγματική γλώσσα είναι ρουτίνα χωρίς παραπάνω σκέψη.

Διάβασε X

Αν $X > 5$ τότε

$\psi \leftarrow X - 3$

Τέλος_αν

Εμφάνισε ψ

Δεν θα πρέπει να υπάρχουν εντολές που να υπάρχει αμφιβολία για τον τρόπο εκτέλεσης τους
Αν το ψ δεν πάρει τιμή, είναι απροσδιόριστο το πώς θα εκτελεστεί η εντολή Εμφάνισε

Μια εντολή δεν ικανοποιεί το κριτήριο της καθοριστικότητας όταν εκτελείται για ένα σύνολο τιμών, αλλά δεν εκτελείται για κάποια τιμή.

Διάβασε α, β, γ

$\gamma \leftarrow \alpha / (\beta - 2)$

Εμφάνισε ψ

Παραβίαση καθοριστικότητας. Αν το β πάρει την τιμή δύο τότε έχουμε διαίρεση με το μηδέν. Το οποίο είναι αδύνατο. Ο αλγόριθμος δεν εκτελείται

Αποτελεσματικότητα

Αποτελεσματικότητα είναι το να είναι η κάθε εντολή αρκετά απλή έτσι ώστε να μπορεί να εκτελεστεί από τον υπολογιστή. Δηλαδή θα πρέπει να υπάρχει εντολή αντίστοιχη στη γλώσσα προγραμματισμού έτσι ώστε να μπορεί ο αλγόριθμος να μετατραπεί σε εκτελέσιμο πρόγραμμα.

Παράδειγμα παραβίασης αυτού του κριτηρίου είναι η εντολή «Βρες το μεγαλύτερο από αυτούς τους 100 αριθμούς». Οι γλώσσες προγραμματισμού (του επιπέδου που μιλάμε) δεν έχουν τέτοια εντολή. Χρειάζεται σύνολο εντολών για να βρούμε το μεγαλύτερο από 100 αριθμούς.

Τέτοιου είδους παραβιάσεις οδηγούν σε μη εκτελέσιμο αλγόριθμο.

Με βάση τα παραπάνω φαίνεται η δεύτερη περίπτωση της καθοριστικότητας και η αποτελεσματικότητα έχουν ένα κοινό: η απουσία και των 2 οδηγούν σε μη εκτελέσιμο αλγόριθμο.

Η διαφορά είναι η αιτία η οποία βρίσκεται πίσω από τη μη εκτελεσιμότητα. Στην περίπτωση 2 της καθοριστικότητας η αιτία είναι η ασάφεια δηλαδή το ότι δεν είναι καλά καθορισμένο κάποιο βήμα.

Στην αποτελεσματικότητα η αιτία είναι η έλλειψη απλότητας στην εντολή δηλαδή η εντολή δεν είναι αρκετά απλή για να γίνει κατανοητή από αυτόν που εκτελεί τον αλγόριθμο.

Μια εντολή δεν ικανοποιεί το κριτήριο της αποτελεσματικότητας όταν δεν μπορεί να εκτελεστεί καθόλου

1. **Βρες_το_μεγαλύτερο_αριθμό**
 2. **$\beta \leftarrow \alpha / 0$, ο διαιρέτης πρέπει να είναι μη μηδενικός**
 3. **$\gamma \leftarrow \alpha \gamma *$, αντί για $\gamma \leftarrow \alpha * \gamma$**
 4. **$\alpha \leftarrow T_P(-1)$, το όρισμα της συνάρτησης T_P πρέπει να είναι μη αρνητικός αριθμός**
- $\alpha \leftarrow T_P(\#)$, το όρισμα της συνάρτησης T_P πρέπει να είναι μη αρνητικός αριθμός**

Περατότητα

Είναι δυνατόν να μεταβάλλεται η μεταβλητή της συνθήκης και ο βρόχος να είναι ατέρμων.

Για παράδειγμα:

$\chi \leftarrow -1$

Αρχή_επανάληψης (Επανάλαβε στην ψευδογλώσσα)

$\chi \leftarrow \chi + 2$

Μέχρις_ότου($\chi = 10$)

Το χ μεταβάλλεται εντός του βρόχου αλλά δεν παίρνει ποτέ την τιμή 10...