

Ανάπτυξη και Σχεδίαση Λογισμικού

Η γλώσσα
προγραμματισμού C

Γεώργιος Δημητρίου

Πίνακες και Βρόχοι

- Πίνακες
 - Δομές δεδομένων που ορίζονται ως διαδοχικά αποθηκευμένα στοιχεία ίδιου τύπου
 - Ο σημαντικότερος σύνθετος τύπος στις περισσότερες γλώσσες προγραμματισμού
- Βρόχοι
 - Προγραμματιστικές δομές επανάληψης
 - Ιδανικοί για επεξεργασία πινάκων
 - Δεν νοείται πρόγραμμα χωρίς κάποιο βρόχο!

Πίνακες της C

- ◉ Δήλωση πινάκων
- ◉ Διαστάσεις πινάκων
- ◉ Αρχικοποίηση πινάκων
- ◉ Αναφορά σε στοιχεία πίνακα
 - ◉ Έλεγχος (ή μη έλεγχος) ορίων πίνακα!

Δήλωση Πινάκων

- Ειδικός προσδιοριστής τροποποίησης τύπου (modifier) μέσω του τελεστή '['

```
int a[100];
```

- Ο προσδιοριστής [] επιδρά μόνο στο τρέχον όνομα μεταβλητής

```
int a[100], b, c[10];
```

Διαστάσεις

- ◉ Άμεσα μόνο μονοδιάστατοι
 - ◉ Ο προσδιοριστής [] ορίζει μόνο μία διάσταση
- ◉ Έμμεσα και πολυδιάστατοι
 - ◉ Πολλαπλοί προσδιοριστές []
 - ◉ Στοιχεία τύπου άλλου πίνακα
 - ◉ Μετά από δήλωση typedef

Αρχικοποίηση Πινάκων

- Τιμές σε άγκιστρα

```
int a[10] = {0, -1, 2, 3, 0, 0, 5, -2, 1, 0};
```

```
int e[2][3] = {{0, -2, 1}, {1, -1, 0}};
```

```
int b[] = {3, 2}, c, d[4] = {0, -1};
```

```
int f[2][3] = {0, -2, 1, 1, -1, 0};
```

Αναφορά σε Στοιχείο Πίνακα

- Τα στοιχεία ενός πίνακα αριθμούνται από 0 μέχρι το μέγεθος του πίνακα μείον 1.
- Αναφορά με τη βοήθεια του τελεστή '['']'

καποιος_pinakas [expr]

a[0] = a[1] + 2;

b[i][j+1] = c[0] * d[i] + j;

scanf ("%d", &a[k]);

έκφραση δεικτη (index)

Ανάπτυξη και Σχεδίαση Λογισμικού
Η γλώσσα προγραμματισμού C

Έλεγχος Ορίων Πίνακα

- Πώς μπορούμε να είμαστε πάντα βέβαιοι ότι η μεταβλητή i της έκφρασης $a[i]$ έχει τιμή μεταξύ 0 και του μεγέθους του πίνακα;
 - ΔΕΝ μπορούμε!
 - ΔΕΝ ενδιαφέρει τη C!
- Άρα ο προγραμματιστής πρέπει να γράφει κώδικα ελέγχου ώστε να μην ξεφεύγει από τα όρια του πίνακα...

Η Δήλωση typedef

- Συνώνυμα τύπων

```
typedef int array_of_10_integers[10];
```

```
array_of_10_integers x, y = {0, 1}, z;
```

```
array_of_10_integers w[100];
```

```
typedef int new_int;
```

Εντολές Επανάληψης

- ◉ Δημιουργία βρόχων
 - ◉ Μετρητής επαναλήψεων
 - ◉ Έλεγχος τερματισμού

- ◉ Ιδανικές για επεξεργασία πινάκων

Εντολή while

- Όπως την είδαμε στο προηγούμενο μάθημα

```
x = 0;
while (x < 10) {
    x++;
    if (x <= 5) i += 2;
    else i -= 1;
}
```

} σώμα του βρόχου

Εντολή do-while

- Μορφή:

`do stmt while (expr);`

- που σημαίνει:

Να εκτελείς την εντολή `stmt` μέχρι η έκφραση `expr` να γίνει ψευδής,

Άρα η εντολή `stmt` εκτελείται τουλάχιστον μία φορά!

Εντολή do-while

```
do {  
    i = (i + 1) % 3;  
    switch (i) {  
        case 0: a[i]++; break;  
        case 1: a[i] -= 3;  
        case 2: break;  
    }  
}  
while (a[0] + a[1] + a[2] > 0);
```

Εντολή for

- Μορφή:

`for (expr1; expr2; expr3) stmt`

- που σημαίνει:

Πρώτα αποτίμησε την έκφραση `expr1`

Όσο η έκφραση `expr2` είναι αληθής να εκτελείς την εντολή `stmt` ακολουθούμενη από αποτίμηση της έκφρασης `expr3`

Μορφή Εντολής for

- Τυπικά η πρώτη έκφραση υλοποιεί την αρχικοποίηση κάποιου μετρητή επαναλήψεων
- Η δεύτερη έκφραση υλοποιεί τον έλεγχο του βρόχου
- Η τρίτη έκφραση υλοποιεί τη μεταβολή του μετρητή
- Πολλαπλές εκφράσεις μπορούν να συνενώνονται σε μία έκφραση με τη βοήθεια του τελεστή ‘,’

Εντολή for

```
for (i = 0, k = 10; (a[i] != 0) && (k > 2); i++, k--)  
    c[k - i] = b[k]/a[i];
```



```
i = 0;  
k = 10;  
while ((a[i] != 0) && (k > 2)) {  
    c[k - i] = b[k]/a[i];  
    i++;  
    k--;  
}
```


Φωλιασμένοι Βρόχοι

- Μια εντολή επανάληψης μπορεί να βρίσκεται φωλιασμένη μέσα σε άλλη εντολή επανάληψης
- Ιδανική δομή για πολυδιάστατους πίνακες ή πολύπλοκους αλγορίθμους

Η εντολή break

- Πρόωρος τερματισμός του βρόχου στον οποίο συναντάται
 - Όχι τερματισμός εξωτερικού βρόχου!
- Προσοχή αν είναι μέσα σε εντολή switch η οποία είναι φωλιασμένη στο βρόχο!
- Αν η break δεν ενεργεί στο βρόχο που θέλουμε, τότε ο πρόωρος τερματισμός πρέπει να κωδικοποιηθεί με **άλλον τρόπο...**

Η εντολή `continue`

- Συνέχισε με τον έλεγχο τερματισμού και την επόμενη επανάληψη!
- Ειδικά στην εντολή `for`, συνέχισε με την αποτίμηση της `expr3` και μετά πήγαινε στον έλεγχο τερματισμού και την επόμενη επανάληψη.

Τελεστής ‘,’

- Ως τελεστής, επιστρέφει την τιμή και τον τύπο του δεξιού τελεστέου

$x = (++a, i);$

- Αποτίμηση πάντα από αριστερά προς τα δεξιά
- Ο τελεστής με τη χαμηλότερη προτεραιότητα!

Βρόχοι και Πίνακες

- Παράδειγμα: Αρχικοποίηση δισδιάστατου πίνακα ακεραίων 100×1000

```
for (i = 0; i < 100; i++)  
    for (j = 0; j < 1000; j++)  
        a[i][j] = 0;
```

Βρόχοι και Πίνακες

- Παράδειγμα: Εύρεση μέγιστου στοιχείου για κάθε γραμμή ενός δισδιάστατου πίνακα ακεραίων 100×1000

```
for (i = 0; i < 100; i++) {  
    max[i] = a[i][0];  
    for (j = 1; j < 1000; j++)  
        if (a[i][j] > max[i])  
            max[i] = a[i][j];  
}
```

Ανάπτυξη και Σχεδίαση Λογισμικού
Η γλώσσα προγραμματισμού C

Βρόχοι και Πίνακες

- Παράδειγμα: Εύρεση θέσης πρώτου μηδενικού στοιχείου για κάθε γραμμή ενός δισδιάστατου πίνακα ακεραίων 100×1000 (-1 αν δεν υπάρχει)

```
for (i = 0; i < 100; i++) {  
    zero[i] = -1;  
    for (j = 0; j < 1000; j++)  
        if (a[i][j] == 0) {  
            zero[i] = j;  
            break;  
        }  
}
```

Ανάπτυξη και Σχεδίαση Λογισμικού
Η γλώσσα προγραμματισμού C

Βρόχοι και Πίνακες

- Παράδειγμα: Εύρεση θέσης πρώτου μηδενικού στοιχείου για κάθε γραμμή ενός δισδιάστατου πίνακα ακεραίων 100×1000 (-1 αν δεν υπάρχει)
– πιο σύντομη υλοποίηση

```
for (i = 0; i < 100; i++) {  
    zero[i] = -1;  
    for (j = 0; j < 1000 && a[i][j] != 0; j++);  
    if (j < 1000 ) zero[i] = j;  
}
```


Τι Μάθαμε Σήμερα

- Πίνακες της C
- Η δήλωση typedef
- Εντολές βρόχου do-while και for
- Εντολές break και continue
- Νέοι τελεστές '[]', ',', '
- Βρόχοι και επεξεργασία πινάκων