



Διάλεξη 4: Δείκτες (pointers) και Πίνακες Δεικτών

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:
Εισαγωγή στις έννοιες:

- Πίνακες Δεικτών
- Πολυδιάστατοι πίνακες
- Πέρασμα παραμέτρων σε προγράμματα C

Διδάσκων: Παναγιώτης Ανδρέου

Πίνακες Δεικτών

Ορισμός Πίνακα με Δείκτες (Pointers): Πίνακας δεικτών είναι ένας πίνακας που περιέχει δείκτες.

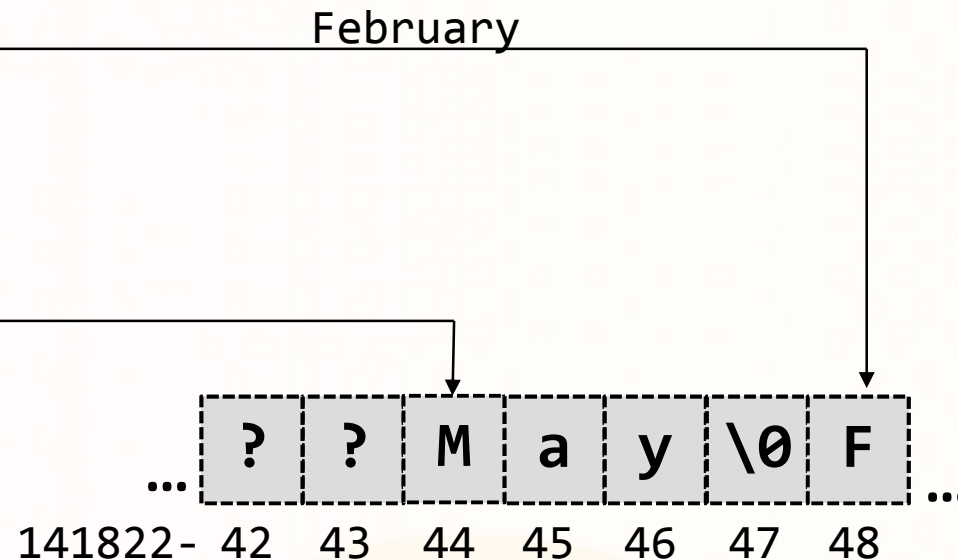
Δηλώνεται με τον **τύπο δεδομένων**, * και [], π.χ., `int * p[]`

- Παράδειγμα: η δήλωση `char *a[10] = {}` λέει ότι ο `a` είναι πίνακας μεγέθους 10 (index: 0-9) και τα στοιχεία του είναι δείκτες σε `char`. Έτσι το `a[6]` είναι δείκτης σε χαρακτήρα και `*a[6]` είναι ο χαρακτήρας στον οποίο δείχνει.
- Αρχικοποίηση τιμών σε πίνακα δεικτών γίνεται με τη σύνταξη.

```
char *m[] = { "January", "February", "March",  
             "April", "May", "June",  
             "July", "August", "September",  
             "October", "November", "December" };
```

Παράδειγμα Πίνακες Δεικτών

	Address	data
m[0]	3143928	14182360
m[1]	3143932	14182248
m[2]	3143936	14182324
m[3]	3143940	14182312
m[4]	3143944	14182244
m[5]	3143948	14182236
m[6]	3143952	14182204
m[7]	3143956	14182228
m[8]	3143960	14182216
m[9]	3143964	14182300
m[10]	3143968	14182288
m[11]	3143972	14182276



Οι χαρακτήρες κάθε συμβολοσειράς τοποθετούνται κάπου στη μνήμη και στο m[i] τοποθετείται δείκτης σ' αυτούς τους χαρακτήρες.

Παράδειγμα Πίνακες Δεικτών (συν.)

	Address	data	
m[0]	3143928	14182360	January
m[1]	3143932	14182248	February
m[2]	3143936	14182324	March
m[3]	3143940	14182312	April
m[4]	3143944	14182244	May
m[5]	3143948	14182236	June
m[6]	3143952	14182204	July
m[7]	3143956	14182228	August
m[8]	3143960	14182216	September
m[9]	3143964	14182300	October
m[10]	3143968	14182288	November
m[11]	3143972	14182276	December

Τι θα τυπώσουν τα ακόλουθα;

1. `printf("%c", *m[1]);`
2. `printf("%c", *(m[3]+2));`
3. `printf("%s", m[0]);`
4. `printf("%d", m[0]);`
5. `printf("%d", &m[9]);`
6. `printf("%c", m[2][3]);`

1. F
2. r
3. January
4. 14182360
5. 3143964
6. c

Παράδειγμα Πίνακες Δεικτών (συν.)

- Πρόβλημα: Χρειαζόμαστε μία συνάρτηση η οποία θα επιστρέφει το ένα string με το όνομα του κάποιου μήνα k που θα δοθεί σαν παράμετρος
 - Παράμετροι Εισόδου:
 - `char *months[]`: πίνακας δεικτών με τους μήνες
 - `int k`: ο αριθμός του μήνα (π.χ., 1=January)

```
char *month_name(char *months[], int k){  
    if (k < 1 || k > 12)  
        return "Error";  
    else  
        return months[k-1];  
}
```

Αρχικοποίηση Πίνακα Δεικτών

```
int main() {  
  
    int *a[10];  
    for(int i=0; i<10; i++){  
        printf("%d\n", a[i]);  
    }  
}
```

```
int *b[10] = {};  
for(int i=0; i<10; i++){  
    printf("%d\n", b[i]);  
}  
}
```

ΕΚΤΥΠΩΝΕΙ

```
1628344792  
2289544  
2088999592  
2088773112  
-1  
2088773104  
2088772914  
2089866642  
2088803211  
1892
```

ΕΚΤΥΠΩΝΕΙ

```
0  
0  
0  
0  
0  
0  
0  
0  
0  
0  
0
```

ΠΡΟΣΟΧΗ

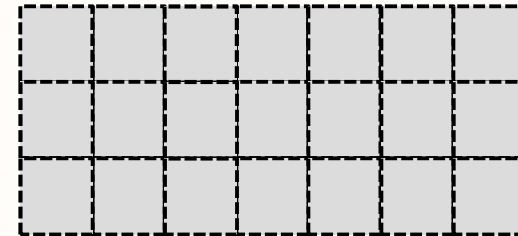
Τον χώρο στον οποίο θα δείχνει κάθε pointer $a[i]$, πρέπει να τον δεσμεύσουμε ξεχωριστά

Πολυδιάστατοι πίνακες

- Η C διαθέτει ορθογωνικούς πολυδιάστατους πίνακες.
- Κατά τη δήλωση ενός πολυδιάστατου πίνακα (όπως και για ένα μονοδιάστατο) πρέπει να προσδιορίσουμε τις διαστάσεις του*.

π.χ., **τύπος ονομα[γραμμες][στήλες]**

```
char array[ 3 ][ 7 ]
```



- Ένας πολυδιάστατος πίνακας είναι ένας μονοδιάστατος πίνακας κάθε στοιχείο του οποίου είναι ένας πίνακας.
- Για να αναφερθούμε σε κάποιο στοιχείο του πίνακα χρησιμοποιούμε δείκτες θέσης.

π.χ., **array[i][j]** **/*[γραμμή][στήλη]*/**

- * Στην C99 επιτρέπονται και οι δυναμικοί πίνακες, π.χ., *char array[i][i]*, όπου *i* είναι μεταβλητή αλλά δεν θα τους χρησιμοποιήσουμε στα πλαίσια αυτού του μαθήματος.

Πολυδιάστατοι πίνακες (συν.)

- Αρχικοποίηση πολυδιάστατου πίνακα

Μια λίστα αρχικών τιμών κλεισμένη σε άγκιστρα, όπου κάθε τιμή παίρνει αρχική τιμή από μια αντίστοιχη υπολίστα.

π.χ.

```
char daysofmonth[2][13] = {  
    {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},  
    {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}}
```

- Μεταβίβαση δισδιάστατου πίνακα σε συνάρτηση

Η δήλωση των παραμέτρων της συνάρτησης **πρέπει να περιλαμβάνει τον αριθμό των στηλών** κάθε δισδιάστατου πίνακα. Ο αριθμός των γραμμών είναι αδιάφορος.

Για παράδειγμα, όλες οι πιο κάτω δηλώσεις είναι έγκυρες.

```
f( int daysofmonth[2][13] ) { . . . }  
f( int daysofmonth[][13] ) { . . . }  
f( int(*daysofmonth)[13] ) { . . . }
```


Πολυδιάστατοι Πίνακες vs. Πίνακες Δεικτών

- Έστω

```
int a[10][20];  
int *b[10];
```

- Ποια η διαφορά ανάμεσα στους δύο πίνακες;
 - Ο a είναι πραγματικά δισδιάστατος πίνακας: κατά τον ορισμό του δεσμεύθηκαν $10 \times 20 = 200$ συνεχόμενες θέσεις.
 - Κατά τον ορισμό του b κατανέμεται χώρος για 10 δείκτες. Απόδοση αρχικών τιμών πρέπει να γίνει ρητά είτε στατικά (i.e. { }) ή δυναμικά με κώδικα (for loop).

```
int main() {  
    int a[10][20];  
    int *b[10];  
    printf("%d, %d", sizeof(a),  
    sizeof(b));  
}
```

- Εκτυπώνει 800 (10 γραμμές x 20 στήλες x 4 (μέγεθος int) και 40 (10 γραμμές x 4 (μέγεθος pointer)) σε **x86**

- Εκτυπώνει 800 (10 γραμμές x 20 στήλες x 4 (μέγεθος int) και 80 (10 γραμμές x 8 (μέγεθος pointer)) σε **x64**

- **Πλεονέκτημα Πίνακα Δεικτών**

Κάθε δείκτης μπορεί να δείχνει σε **γραμμή με διαφορετικό μήκος**

Πολυδιάστατοι Πίνακες vs. Πίνακες Δεικτών (συν.)

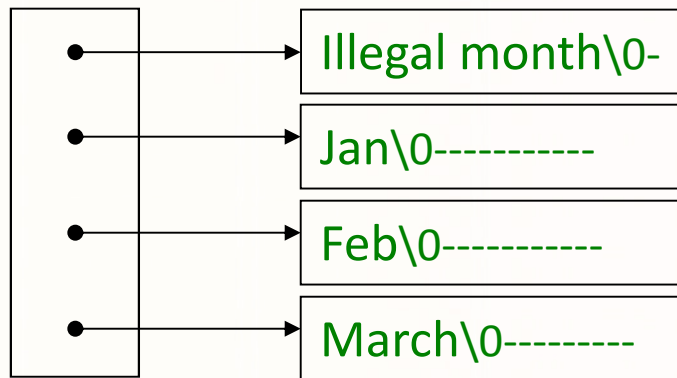
Παράδειγμα

Έστω

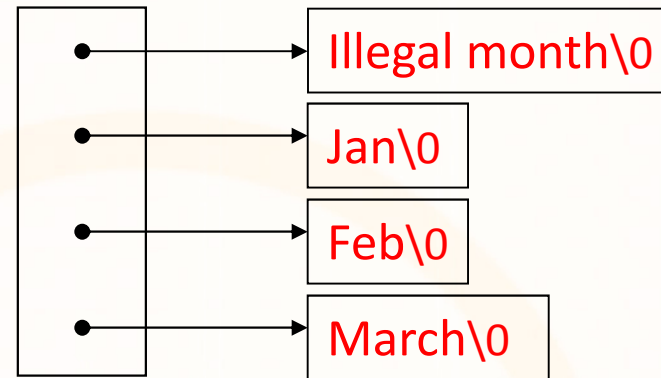
```
char fixed[] [15] = {"Illegal month", "Jan", "Feb", "March"};
```

```
char *variable[] = {"Illegal month", "Jan", "Feb", "March"};
```

Γραφικά στο επίπεδο της μνήμης έχουμε



fixed



variable

Παράμετροι της συνάρτησης <main>

- Η συνάρτηση **main** με παράμετρο **void** θέλοντας να δείξουμε ότι η συνάρτηση **main** δε δέχεται ορίσματα.

```
int main() { ... }
```

- Αυτό όμως δε σημαίνει ότι δεν μπορούμε να περάσουμε ορίσματα. Τα δυνατά ορίσματα όμως της **main** είναι καθορισμένα και είναι τα εξής:

```
int main(int argc, char *argv[]) { ... }
```

- Τα ορίσματα περνιούνται στο πρόγραμμα από τη γραμμή εντολών τη στιγμή που αρχίζει η εκτέλεσή του.
- Το πρώτο όρισμα **argc**, το οποίο είναι τύπου ακέραιος, είναι ο αριθμός των ορισμάτων της γραμμής διαταγών, με τα οποία έχει κληθεί το πρόγραμμα.
- Παράδειγμα: \$ prog opt1 opt2 opt3 **argc=4**
- Παράδειγμα: \$ prog 13 **argc=2**

Παράμετροι της συνάρτησης <main> (συν.)

- Το δεύτερο όρισμα **argv** είναι δείκτης για έναν πίνακα συμβολοσειρών ο οποίος περιέχει τα ορίσματα. Κατά σύμβαση
 - **argv[0]** είναι το όνομα με το οποίο κλήθηκε το πρόγραμμα.
 - Παράδειγμα: \$ prog1 **argv[0]=prog1**
 - Παράδειγμα: \$ prog2 **argv[0]=prog2**
 - **argv[1], ..., argv[argc - 1]**, είναι τα υπόλοιπα ορίσματα με την σειρά που δόθηκαν στη γραμμή εντολής.
 - Παράδειγμα: \$ prog opt1 opt2 opt3
 - **argc=4**
 - **argv[0]=prog, argv[1]=opt1, argv[2]=opt2, argv[3]=opt3**
 - Παράδειγμα: \$ prog 13
 - **argc=2**
 - **argv[0]=prog, argv[1]=13**