Διάλεξη 4: Δείκτες και Πίνακες

Στην ενότητα αυτή θα μελετηθούν τα εξής θέματα:

Πίνακες Δεικτών, Παραδείγματα,

Πολυδιάστατοι πίνακες

Πέρασμα παραμέτρων σε προγράμματα C

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

Πίνακες Δεικτών

- Πίνακας δεικτών είναι ένας πίνακας που περιέχει δείκτες.
- Δήλωση πίνακα από δείκτες τύπου type γίνεται ως

  type *arrayname[arraysize]

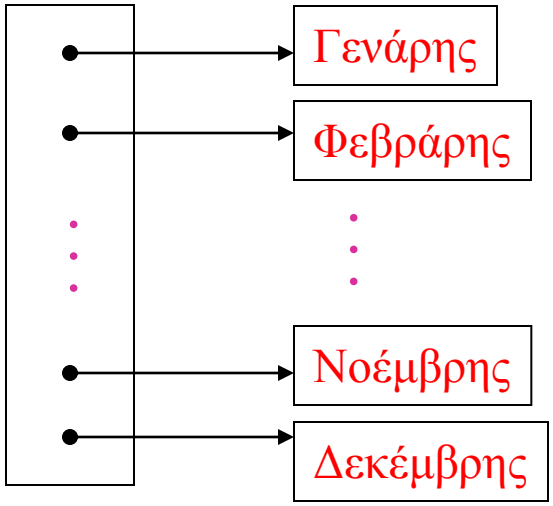
  π.χ. η δήλωση char *a[10] = { } λέει ότι ο a είναι πίνακας μεγέθους 10 και τα στοιχεία του είναι δείκτες σε char. Έτσι το a[6] είναι δείκτης σε χαρακτήρα και *a[6] είναι ο χαρακτήρας στον οποίο δείχνει.
- Αρχικοποίηση τιμής σε πίνακα δεικτών γίνεται με τη γνωστή σύνταξη.
- Παράδειγμα:

```
char *name[]  = {
        "Γενάρης", "Φεβράρης",  "Μάρτης", "Απρίλης",
        "Μάης", "Ιούνης", "Ιούλης", "Αύγουστος",
        "Σεπτέμβρης", "Οκτώβρης", "Νοέμβρης", "Δεκέμβρης"}
```

Παράδειγμα

```
char *name[]  = {
            "Γενάρης", "Φεβράρης",  "Μάρτης", "Απρίλης",
            "Μάης", "Ιούνης", "Ιούλης", "Αύγουστος",
            "Σεπτέμβρης", "Οκτώβρης", "Νοέμβρης", "Δεκέμβρης"}
```

- Γραφικά, έχουμε



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

Παράδειγμα

- Να γράψετε συνάρτηση που, με δεδομένο εισόδου ακέραιο i, επιστρέφει το όνομα του i-οστού μήνα (αν υπάρχει).

```
#include <stdio.h>

 char *month_name(char *name[], int n){
       if (n < 1 || n > 12)
             return name[0];
       else
             return name[n];
  }

int main() {
      char *name[]  = { "Ανύπαρκτος μήνας","Γενάρης", "Φεβράρης",
      "Μάρτης", "Απρίλης", "Μάης", "Ιούνης", "Ιούλης", "Αύγουστος",
      "Σεπτέμβρης", "Οκτώβρης", "Νοέμβρης", "Δεκέμβρης"};

      printf("%s", month_name(name, 1));
      return 0;
}
```

Σωστή Αρχικοποίηση

```
#include <stdio.h>

int main() {
        char  *a[10] ;
        int i;

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

Εκτυπώνει

1628344792
2289544
2088999592
2088773112
-1
2088773104
2088772914
2089866642
2088803211
1892

```
#include <stdio.h>

int main() {
        char  *a[10] = { };
        int i;

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

Εκτυπώνει

0
0
0
0
0
0
0
0
0
0

Προσοχή: Τον χώρο στον οποίο θα δείχνει κάθε pointer a[i], πρέπει να τον δεσμεύσουμε ξεχωριστά (i.e. name[])

```
#include <stdio.h>

int main() {
                char  *a[10] = {};
                char name[] = “car";

                a[0] = name;

                printf("%s\n", a[0]);
}
```

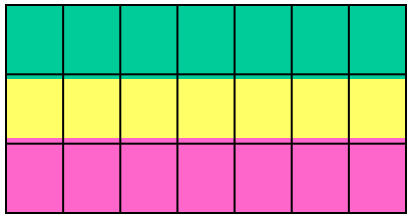
Εκτυπώνει «car»

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.

  ```
  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. *γραμμές* *στήλες*

```
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 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 (10x20x4) και 40(10x4) σε x86
Εκτυπώνει 800 (10x20x4) και 80(10x8) σε x64

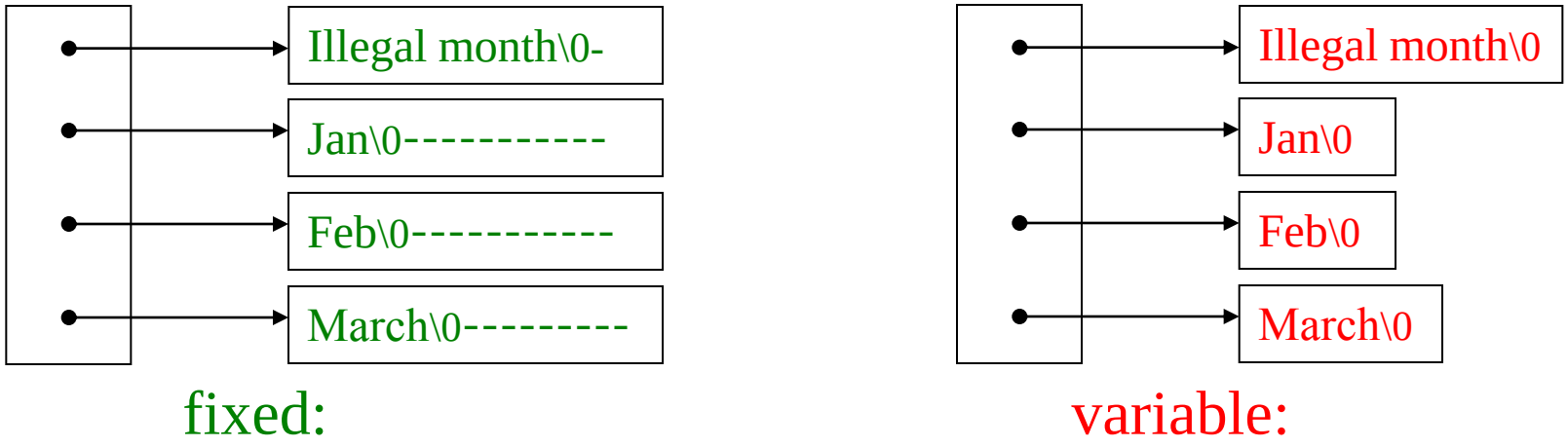
- **Πλεονέκτημα Πίνακα Δεικτών**
  Κάθε δείκτης μπορεί να δείχνει σε **γραμμή με διαφορετικό μήκος**

Παράδειγμα

Έστω

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

Γραφικά στο επίπεδο της μνήμης έχουμε



- συμβολίζει ένα αχρησιμοποίητο byte

Παράμετροι της συνάρτησης main()

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

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

- 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.

Σημασία των ορισμάτων

- Το πρώτο όρισμα argc, το οποίο είναι τύπου ακέραιος, είναι ο αριθμός των ορισμάτων της γραμμής διαταγών, με τα οποία έχει κληθεί το πρόγραμμα.

- 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.
  - argv[1], …, argv[argc - 1], είναι τα υπόλοιπα ορίσματα με την σειρά που δόθηκαν στη γραμμή εντολής.
  - argv[argc] περιέχει το μηδενικό δείκτη (NULL).

Σημασία των ορισμάτων (συν.)

- Παράδειγμα: Έστω ένα πρόγραμμα C το οποίο έχει τη μορφή:

```
int main(int argc, char *argv[]) {
        int i = 0;
        for (i=0; i<argc; i++)  {
              printf("argv[%d] = \"%s\"\n", i, argv[i]);
        }
        return 0;
}
```

- Θεωρείστε επίσης ότι το εκτελέσιμο αρχείο του παραπάνω προγράμματος έχει ονομαστεί prog. Τότε κατά την κλήση του prog υπό τη μορφή:

```
$ prog  opt1  opt2  opt3
```

έχουμε την εξής ανάθεση τιμών στα ορίσματα της main:

argc = 4

argv[0] = "prog"
argv[1] = "opt1"
argv[2] = "opt2"
argv[3] = "opt3"

Παράδειγμα 1

- Ζητούμενο: πρόγραμμα που κατά την κλήση του με n ορίσματα αντηχεί (echoes) τα n-1 τελευταία στην οθόνη. Για παράδειγμα αν το πρόγραμμα αυτό λέγεται echo, τότε μία κλήση της μορφής:

  $ echo Hello world!

  θα εμφάνιζε στην έξοδο

  $ Hello world!

- Οι τυπικές παράμετροι της συνάρτησης main θα έχουν τιμές:

  argc = 3

  argv[0] = "echo"
  argv[1] = "Hello"
  argv[2] = "world!"

Παράδειγμα 1 - Υλοποίηση (συν.)

```
#include <stdio.h>

main (int argc, char *argv[]){
    int i;
    for(i = 1; i < argc; i++)
        printf("%s%s", argv[i],
                        (i < argc - 1) ? " " : "");
    printf("\n");
    return 0;
}
```

Μακροεντολή C (Macro)

```
(i < argc - 1) ? " " : "");
```

← Ισοδύναμο με →

```
If (i < argc - 1)
    return " "
Else
    return ""
```