

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΕΣ ΤΕΧΝΙΚΕΣ

<http://www.softlab.ntua.gr/~nickie/Courses/progtech/>

Διδάσκοντες: Γιάννης Μαΐστρος (maistros@cs.ntua.gr)
Στάθης Ζάχος (zachos@cs.ntua.gr)
Νίκος Παπασπύρου (nickie@softlab.ntua.gr)

Διαφάνειες παρουσίασης #5 (α)

- ✓ Δομές και ενώσεις
- ✓ Δείκτες
- ✓ Πίνακες και δείκτες

Δομές (structures)

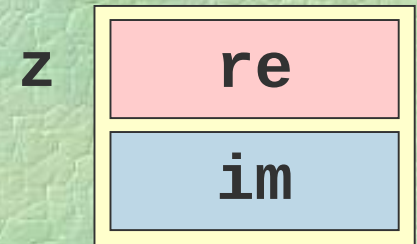
(i)

◆ Ο τύπος struct

```
struct complex_tag {  
    double re, im;  
};
```

```
struct complex_tag z;  
z.re = 1.0;  
z.im = -1.0;
```

```
typedef struct complex_tag complex;  
complex z;  
complex zs [100];
```



◆ Παράδειγμα

```
complex complement (complex z)
{
    complex result;

    result.re = z.re;
    result.im = -z.im
    return result;
}
```


Δομές (structures)

(iii)

◆ Παράδειγμα

```
typedef struct {  
    char    firstName [30];  
    char    lastName  [50];  
    char    phone     [10];  
    int     age;  
    double  salary;  
} record;
```

```
record r [1000];  
r[423].age = 32;
```


Ενώσεις (unions)

◆ Ο τύπος union

```
union number_tag {  
    int i;  
    double d;  
};
```

```
union number_tag n;
```

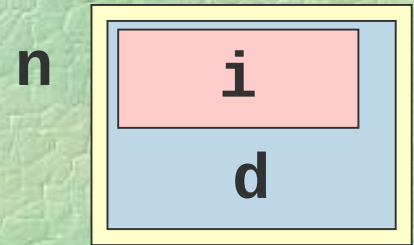
```
n.d = 1.2;
```

```
printf("%lf\n", n.d);
```

```
n.i = 42;
```

```
printf("%d\n", n.i);
```

```
printf("%lf\n", n.d);    /* wrong! */
```



Δείκτες (pointers)

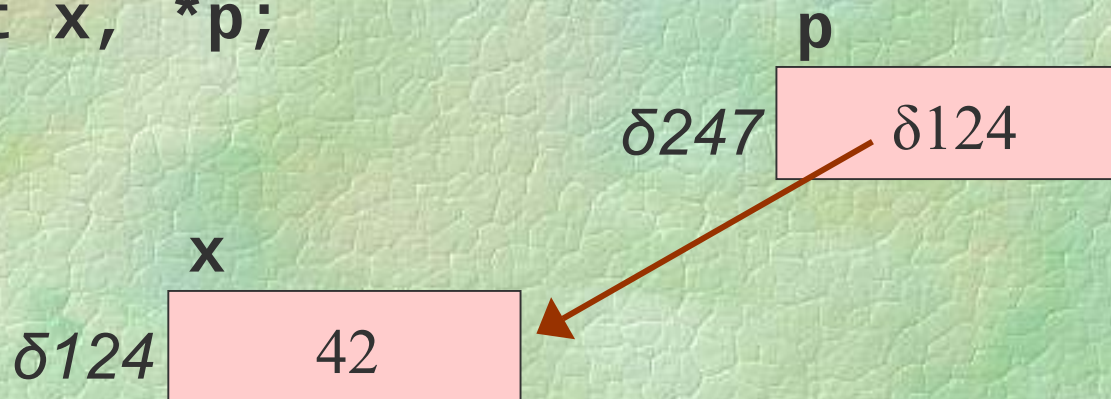
(i)

◆ Μεταβλητές

- που δεν περιέχουν μια τιμή κάποιου απλού τύπου δεδομένων
- αλλά τη **διεύθυνση** μιας άλλης μεταβλητής

◆ Παράδειγμα

```
int x, *p;
```



Δείκτες (pointers)

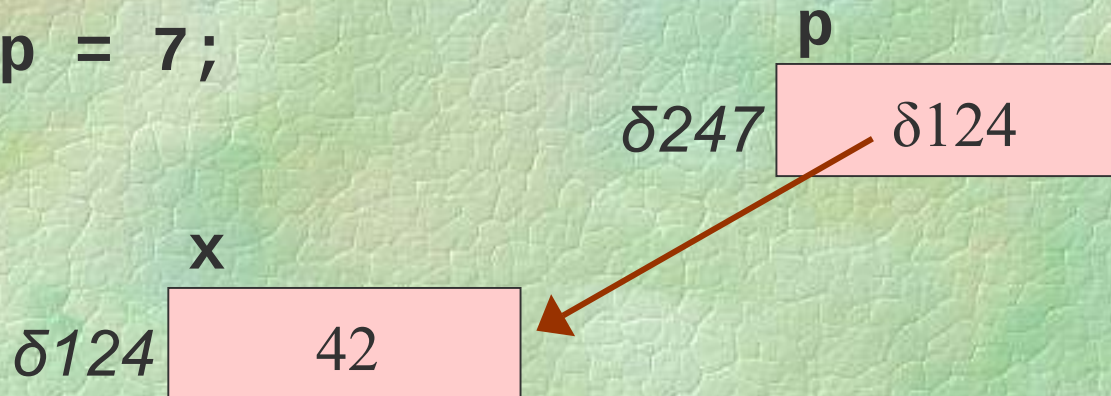
(ii)

◆ Δεικτοδότηση: &

- η διεύθυνση μιας μεταβλητής
`p = &x;`

◆ Αποδεικτοδότηση: *

- το περιεχόμενο μιας διεύθυνσης
`printf("%d", *p);`
`*p = 7;`



Πίνακες και δείκτες

(i)

◆ Αριθμητική δεικτών

```
int a[3] = {7, 6, 42};
```

```
int *p;
```

```
p = &(a[0]);
```

```
p = &a;
```

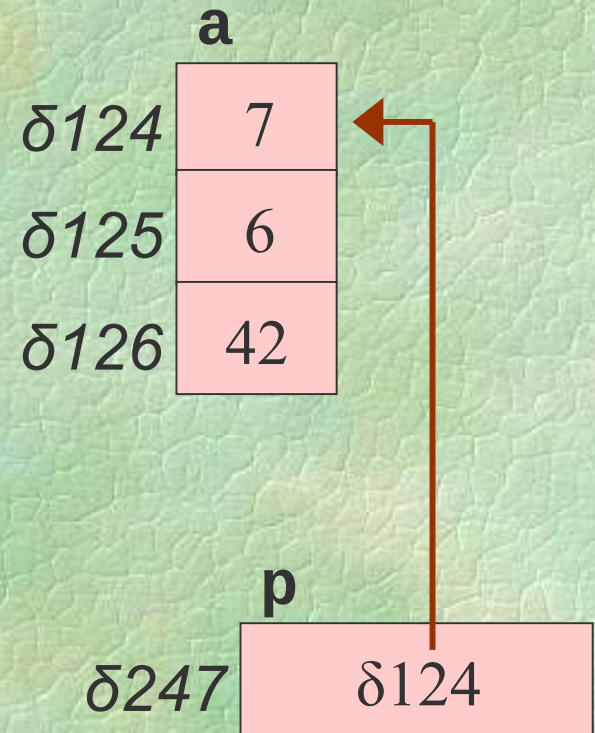
```
p = a;
```

```
printf("%d\n", *p);
```

```
printf("%d\n", *(p+1));
```

```
p = p+2;
```

```
printf("%d\n", *p);
```



◆ Ισοδυναμία πινάκων και δεικτών

- Ένας πίνακας είναι ένας δείκτης στο πρώτο στοιχείο.
- $a[i]$ ισοδύναμο με $*(a+i)$
- Οι πίνακες όμως είναι σταθεροί δείκτες, δηλαδή δεν μπορούν να αλλάξουν τιμή

```
int a[3] = {7, 6, 42};  
int *p = &a;  
p++; /* correct */  
a++; /* wrong! */
```


◆ Παράδειγμα: αντιγραφή πινάκων

```
int a[10], b[10];  
int *p = a, *q = b, i;  
  
/* assume b initialized */  
  
for (i=0; i<10; i++)  
    *p++ = *q++;
```


◆ Παράδειγμα: εκτύπωση συμβολοσειράς

```
void putchar (char c);
```

```
char s[] = "Hello world!\n";
```

```
char *p;
```

```
for (p = s; *p != '\0'; p++)  
    putchar(*p);
```