

ΕΠΛ232 – Προγραμματιστικές Τεχνικές και Εργαλεία

Δείκτες και Συναρτήσεις (Κεφάλαιο 11, ΚΝΚ-2ΕΔ)

Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

<http://www.cs.ucy.ac.cy/courses/EPL232>

Παράδειγμα 1

- Έστω το ακόλουθο πρόγραμμα, το οποίο δείχνει μόνο δηλώσεις συναρτήσεων και μεταβλητών

```
int b, c;
void f(void)
{
    int b, d;
}
void g(int a)
{
    int c;
    {
        int a, d;
    }
}
int main(void)
{
    int c, d;
}
```

- Ποιο το εύρος εμβέλειας της κάθε συναρτήσεως που αναφέρονται πιο κάτω. Αν υπάρχουν περισσότερες από μια παράμετροι με το ίδιο όνομα μέσα στην συνάρτηση, να αναφέρετε ποια είναι ορατά.
 - Η συνάρτηση `f`
 - Η συνάρτηση `g`
 - Το μπλοκ όπου δηλώνονται οι `a`, `d`
 - Η συνάρτηση `main`



Παράδειγμα 1

- Οι b , c , και d είναι ορατές. Η τοπική b κρύβει την εξωτερική b .
- Οι a , b , και c είναι ορατές. Η τοπική c κρύβει την εξωτερική c .
- Οι a , b , c , και d είναι ορατές. Το μπλοκ a κρύβει την παράμετρο a .
- Οι b , c , και d είναι ορατές. Η τοπική c κρύβει την εξωτερική c .



Παράδειγμα 3

- Τι πάει λάθος με το πιο κάτω πρόγραμμα (σκοπός του είναι να βρει το άθροισμα και μέσο όρο των αριθμών του πίνακα a με μέγεθος n);

```
void avg_sum(double a[], int n, double *avg, double *sum)
{
    int i;

    sum = 0.0;
    for (i = 0; i < n; i++)
        sum += a[i];
    avg = sum / n;
}
```

Παράδειγμα 3

- Λείπουν οι τελεστές *:

```
void avg_sum(double a[], int n, double *avg, double *sum)
{
    int i;

    *sum = 0.0;
    for (i = 0; i < n; i++)
        *sum += a[i];
    *avg = *sum / n;
}
```



Παράδειγμα 4

Γράψτε μια συνάρτηση

`void split_time(long total_sec, int *hr, int *min, int *sec)`
`total_sec` είναι ο χρόνος σαν συνάρτηση των δευτερολέπτων από τα μεσάνυκτα,
`hr`, `min`, και `sec` είναι δείκτες σε μεταβλητές των οποίων η συνάρτηση αποθηκεύει την αντίστοιχη τιμή σε ώρες, λεπτά, και δευτερόλεπτα.

```
void split_time(long total_sec, int *hr, int *min, int *sec)
{
    *hr = total_sec / 3600;
    *min = total_sec % 3600 / 60;
    *sec = total_sec % 3600 % 60;
}
```



Παράδειγμα 5

Γράψτε μια συνάρτηση

```
int *find_largest(int a[], int n)
```

που θα επιστρέφει τον δείκτη σε ένα στοιχείο του πίνακα *a* με την μεγαλύτερη τιμή



Παράδειγμα 5

```
int *find_largest(int a[], int n)
{
    int i, *largest;

    largest = &a[0];
    for (i = 1; i < n; i++)
        if (a[i] > *largest)
            largest = &a[i];

    return largest;
}
```

Παράδειγμα 6

Γράψτε ένα πρόγραμμα (με χρήση δεικτών) που ζητά από τον χρήστη να δώσει το ποσό σε δολάρια, και μετά εκτυπώνει το μικρότερο απαραίτητο αριθμό από χαρτονομίσματα των \$20, \$10, \$5, \$1.

Π.χ.

```
Enter a dollar amount: 93
```

```
$20 bills: 4
```

```
$10 bills: 1
```

```
 $5 bills: 0
```

```
 $1 bills: 3
```



Παράδειγμα 6

```
#include <stdio.h>
void pay_amount(int dollars, int *twenties, int *tens, int *fives, int *ones);

int main(void)
{
    int amount, twenties, tens, fives, ones;
    printf("Enter a dollar amount: ");
    scanf("%d", &amount);

    pay_amount(amount, &twenties, &tens, &fives, &ones);
    printf("\n");    /* blank line */

    printf("$20 bills: %d\n", twenties);
    printf("$10 bills: %d\n", tens);
    printf(" $5 bills: %d\n", fives);
    printf(" $1 bills: %d\n", ones);
    return 0;
}

void pay_amount(int dollars, int *twenties, int *tens, int *fives, int *ones)
{
    *twenties = dollars / 20;
    dollars -= *twenties * 20;
    *tens = dollars / 10;
    dollars -= *tens * 10;
    *fives = dollars / 5;
    *ones = dollars % 5;
}
```

