

# ΕΠΛ232 – Προγραμματιστικές Τεχνικές και Εργαλεία

Συμβολοσειρές, Δείκτες και Παραδείγματα (Φροντιστήριο)  
Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

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

# Ασκήσεις κατανόησης

- Ποιες από τις παρακάτω κλήσεις είναι λανθασμένες και γιατί;

`printf("%c", '\n');` → **Correct**

`printf("%c", "\n");` → **Incorrect**; use "%c" only to write a char value

`printf("%s", '\n');` → **Incorrect**; use "%s" only to write a string (a value of type char \*)

`printf("%s", "\n");` → **Correct**

`printf('\n');` → **Incorrect**; printf requires a string, not a character

`printf("\n");` → **Correct**

`putchar('\n');` → **Correct**

`putchar("\n");` → **Incorrect**; putchar requires a character, not a string

`puts('\n');` → **Incorrect**; puts requires a string, not a character

`puts("\n");` → **Incorrect**; this writes two new-line characters

`puts("");` → **Correct**



# Ασκήσεις κατανόησης

- Έστω ότι καλούμε την συνάρτηση `scanf` ως ακολούθως:

```
scanf ("%d%s%d", &i, s, &j);
```

Αν ο χρήστης δώσει σαν είσοδο `12abc34 56def78`, τι τιμές θα πάρουν τα `i` (int), `s` (array of characters), και `j` (int);

```
i = 12,
```

```
s = "abc34".
```

```
j = 56.
```



# Ασκήσεις κατανόησης

- Έστω ότι `str` είναι ένα array από χαρακτήρες. Ποια από τις παρακάτω εντολές είναι διαφορετική από τις υπόλοιπες:

(a) `*str = 0;`

(b) `str[0] = '\0';`

(c) `strcpy(str, "");`

(d) `strcat(str, "");`

**(d) is not equivalent to the others.**

(a), (b), and (c) make `str` empty; (d) concatenates an empty string to the end of `str`.



# Ασκήσεις κατανόησης

- Ποια θα είναι η τιμή της συμβολοσειράς  $s1$  όταν καλέσουμε τις ακόλουθες εντολές

```
strcpy(s1, "computer")
strcpy(s2, "science")
if (strcmp(s1,s2)<0)
    strcat(s1,s2)
else
    strcat(s2,s1)
s1[strlen(s1)-6] = '\0'
```

**Απάντηση: "computers"**



# Ασκήσεις κατανόησης

- Τι εκτυπώνει το ακόλουθο πρόγραμμα:

```
#include <stdio.h>
int main(void)
{
    char s[] = "Hsjodi", *p;
    for (p = s; *p; p++)
        --*p;
    puts(s)
    return 0;
}
```

Το πρόγραμμα εκτυπώνει τη λέξη "Grinch", ακολουθούμενη από ένα χαρακτήρα νέας γραμμής. Ο βρόγχος αντικαθιστά κάθε χαρακτήρα του s με τον χαρακτήρα που προηγείται αλφαβητικά.



## Πρόγραμμα

# Έλεγχος του ονόματος των πλανητών

- Το πρόγραμμα `planet.c` απεικονίζει τον τρόπο πρόσβασης σε ορίσματα γραμμής εντολών.
- Το πρόγραμμα έχει σχεδιαστεί για να ελέγξει μια σειρά από συμβολοσειρές κατά πόσο αντιστοιχούν σε ονόματα πλανητών.
- Οι συμβολοσειρές έχουν τεθεί στη γραμμή εντολών:

```
planet Jupiter venus Earth fred
```

- Το πρόγραμμα θα υποδείξει αν η συμβολοσειρά αποτελεί όνομα πλανήτη, και αν είναι, θα εμφανίσει τον αριθμό του πλανήτη:

```
Jupiter is planet 5  
venus is not a planet  
Earth is planet 3  
fred is not a planet
```



# Πρόγραμμα

## Έλεγχος του ονόματος των πλανητών

### planet.c

```
/* Checks planet names */
#include <stdio.h>
#include <string.h>
#define NUM_PLANETS 9
int main(int argc, char *argv[])
{
    char *planets[] = {"Mercury", "Venus", "Earth",
                      "Mars", "Jupiter", "Saturn",
                      "Uranus", "Neptune", "Pluto"};

    int i, j;
    for (i = 1; i < argc; i++) {
        for (j = 0; j < NUM_PLANETS; j++)
            if (strcmp(argv[i], planets[j]) == 0) {
                printf("%s is planet %d\n", argv[i], j + 1);
                break;
            }
        if (j == NUM_PLANETS)
            printf("%s is not a planet\n", argv[i]);
    }
    return 0;
}
```

