

Άσκηση 6.5



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Να γραφεί ένα πρόγραμμα το οποίο θα αντιγράφει ένα αλφαριθμητικό σε ένα νέο αλφαριθμητικό χωρίς να χρησιμοποιηθεί η συνάρτηση `strcpy()`.

Άσκηση 6.5 - main



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

int main() {
    char str1[100], str2[100];
    int i = 0;

    printf("Εισάγετε μια φράση: ");
    gets(str1);
    while (str1[i] != '\0') {
        str2[i] = str1[i];
        i++;
    }
    str2[i] = '\0';
    printf("Η αντιγραμμένη φράση είναι: %s", str2);
    return 0;
}
```

Άσκηση 6.6



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Να γραφεί ένα πρόγραμμα το οποίο θα βρίσκει εάν ένα αλφαριθμητικό είναι παλίνδρομο (δηλ. αν διαβάζεται το ίδιο από την αρχή προς το τέλος και από το τέλος προς την αρχή).

Άσκηση 6.6 - is_palindrome



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#include <string.h>

int is_palindrome(char *str) {

    int length = strlen(str);

    for(int i = 0; i < length/2; i++){
        if(str[i] != str[length-i-1])
            return 0;
    }
    return 1;
}
```

Η σύγκριση **str[i] != str[length - i - 1]** συγκρίνει μεμονωμένα bytes, όχι ολοκληρωμένους ελληνικούς χαρακτήρες.

Αν ο χαρακτήρας αποτελείται από πολλά bytes, τότε ο έλεγχος αποτυγχάνει.

Άσκηση 6.6 - main



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
int main() {
    char input[100];

    printf("Εισάγετε μια λέξη: ");
    gets(input);

    if (is_palindrome(input)) {
        printf("Η λέξη \"%s\" είναι παλίνδρομη!\n", input);
    } else {
        printf("Η λέξη \"%s\" ΔΕΝ είναι παλίνδρομη.\n", input);
    }

    return 0;
}
```

Άσκηση 6.7



Να γραφεί ένα πρόγραμμα το οποίο να προσομοιώνει το δημοφιλές παιχνίδι της κρεμάλας. Το πρόγραμμα να διαβάζει μια λέξη μέχρι 30 χαρακτήρες, η οποία θα αποτελεί τη μυστική λέξη.

Στη συνέχεια, το πρόγραμμα να διαβάζει επαναληπτικά κάποιο γράμμα από τον παίκτη.

- Αν το γράμμα δεν περιέχεται στη μυστική λέξη, το πρόγραμμα να ενημερώνει τον παίκτη.
- Αν περιέχεται, να εμφανίζει τα γράμματα που έχουν βρεθεί και στα υπόλοιπα να βάζει τον χαρακτήρα της υπογράμμισης ("_").

Το παιχνίδι να ολοκληρώνεται όταν βρεθεί η λέξη ή αν ο παίκτης έχει κάνει 7 λανθασμένες προβλέψεις. Θεωρήστε ότι δεν επιτρέπεται ο χρήστης να εισάγει το ίδιο γράμμα πάνω από μια φορά.

Άσκηση 6.7 - main (1/2)



```
#include <stdio.h>
#include <string.h>
#define MAX_ERRORS 7

int main() {
    char secret[31], hide[31], ch;
    int length, found, errors, correct;

    printf("Εισάγετε μια λέξη: ");
    gets(secret);

    length = strlen(secret);

    for(int i = 0; i < length; i++)
        hide[i] = '_';
    hide[length] = '\\0';
```

Άσκηση 6.7 - main (2/2)



```
errors = 0; correct = 0;
while(errors < MAX_ERRORS && correct < length){
    printf("Δώσε έναν χαρακτήρα: ");
    scanf(" %c", &ch);
    found = 0;
    for(int i = 0; i < length; i++){
        if(secret[i] == ch && hide[i]!='_'){
            hide[i] = ch;
            found = 1;
            correct++;
        }
    }
    if(found == 0){
        errors++;
        printf("Το γράμμα %c δεν βρέθηκε\n", ch);
    }
    printf("%s\n",hide);
}
```

```
if(errors == MAX_ERRORS)
    printf("Εχασες\n");
else
    printf("Κέρδισες!\n");

return 0;
}
```