

1. Να γραψετε μια συνάρτηση η οποία υπολογίζει και επιστρέφει ποσοι χαρακτήρες **ch** υπάρχουν σε έναν πίνακα χαρακτήρων με **n** στοιχεία.

```
int syn(char pc[],char ch, int n)
```

Παραλλαγή Με αριθμητική δεικτών

```
int syn(char *pc,char ch, int n)
```

2. Να γραψετε μια συνάρτηση η οποία υπολογίζει και επιστρέφει ποσοι χαρακτήρες **ch** υπάρχουν σε έναν στρίνγκ (πίνακας χαρακτήρων με τερματικό χαρακτήρα).

```
int syn(char s[],char ch)
```

Παραλλαγή Με αριθμητική δεικτών

```
int syn(char *s,char ch)
```

3. Ποιες οι διαφορές μεταξύ άσκησης 1 και άσκησης 2.

4. Να γράψετε μια συνάρτηση η οποία αντιγράφει τους **n** πρώτους χαρακτήρες ενός στρίνγκ **s** σε ένα άλλο στρίνγκ **t**. Στο κυρίως πρόγραμμα δηλώσετε δυο πίνακες χαρακτήρων (strings) **s** και **t** (το στρίνγκ **t** πρέπει να έχει μήκος τουλάχιστον το μήκος του στρίνγκ **s**). Δώσετε τιμή στο στρίνγκ **s** από το τερματικό με **cin**.

```
void cps(char t[], char s[], int n);
```

Παραλλαγή Με αριθμητική δεικτών

```
void cps(char *t, char *s, int n);
```

5. Να γράψετε μια **void** συνάρτηση η οποία αντιγραφεί (χαρακτήρα - χαρακτήρα) ένα στρίνγκ **s** σε ένα άλλο στρίνγκ **t** με την διαφορά ότι στο **t** δεν αντιγράφονται τυχόν προπορευόμενα κενά διαστήματα. Να χρησιμοποιήσετε δείκτες. (Η συνάρτηση **gets** της **C/C++** βιβλιοθήκης διαβάζει και τα κενά διαστήματα)

πχ. το στρίνγκ “ Eva Adam wow “
 γίνεται “Eva Adam wow “

6. Όπως η άσκηση 5 αλλά στο **t** δεν αντιγράφονται οι τελευταίοι κενοί χαρακτήρες

πχ. το στρίνγκ “ Eva Adam wow “
 γίνεται “ Eva Adam wow“

7. Να γράψετε μια συνάρτηση η οποία επιστρέφει έναν **char** δείκτη στην θέση όπου υπάρχει ο χαρακτήρας **ch** (την πρώτη φορά) μέσα σε ένα στρίνγκ **str**. Η συνάρτηση επιστρέφει **NULL** σε περίπτωση που δεν υπάρχει ο χαρακτήρας **ch**. Χρησιμοποιήστε δείκτες για τον πίνακα χαρακτήρων.

```
char* ptoc(char *str, char ch);
```

8. Να γράψετε μια συνάρτηση **TOUPPER** η οποία μετατρέπει ένα πεζό γράμμα (χαρακτήρα) σε κεφαλαίο γράμμα (χαρακτήρα) α) ως **macro** με **#define** β) ως συνάρτηση με το πρόθεμα **inline**.

9. Να γράψετε ένα πρόγραμμα όπου δηλώσετε ένα πίνακα στρινγκς **n** θέσεων **char a[n][100]**, έναν πίνακα **char**-δεικτών με **n** θέσεις **char *pa[n]**, και ένα **char**-δείκτη-σε-δείκτη **char **pp**.

--Καλέσετε μια συνάρτηση με πρωτότυπο **void eis(char [][][100],int n);** όπου δημιουργούνται **n** ονόματα φοιτητών τυχαία (με χαρακτήρες A έως Z) στον πίνακα των στρινγκς. Τα επιθετα των φοιτητών θα έχουν τυχαίο μήκος από 6 έως 20 χαρακτήρες και τα ονόματα μήκος από 4 έως 12 χαρακτήρες. Το επιθετο θα χωρίζεται από το όνομα με ένα κενό διαστήμα (χαρακτήρα space ' ').

--Εμφανίσετε τυχαία 10 ονόματα φοιτητών καλώντας την συνάρτηση με πρωτότυπο **void aon(char *s)** η οποία θα εμφανίζει ένα στρινγκ **s** στην οθόνη.

--Δώσετε τιμές στον πίνακα δεικτών ώστε ο πρώτος δείκτης να δείχνει στο πρώτο στρινγκ, ο δεύτερος στο δεύτερο στρινγκ κοκ. με μια θηλειά **pa[i] = a[i];**

--Καλέσετε μια συνάρτηση με πρωτότυπο **void emf1(char *pa[],int n)** η οποία εμφανίζει τα ονόματα των φοιτητών και το δεύτερο γράμμα του ονόματος.

--Δώσετε τιμή στο δείκτη-σε δείκτη ώστε να δείχνει στο πρώτο στοιχείο του πίνακα δεικτών **pp = pa;**

--Τέλος καλέσετε μια συνάρτηση με πρωτότυπο **void emf2(char **pp,int n);** η οποία εμφανίζει τα ονόματα των φοιτητών και το τρίτο γράμμα του ονόματος