

Άσκηση 1

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει χαρακτήρες και να εμφανίζει πόσοι χαρακτήρες μεταξύ δύο διαδοχικών '\*' είναι: α) γράμματα β) ψηφία και γ) άλλοι χαρακτήρες. Αν δεν υπάρχουν δύο '\*' το πρόγραμμα να εμφανίζει ανάλογο μήνυμα. Αν υπάρχουν περισσότερα των δύο '\*' να λαμβάνονται υπόψη μόνο τα δύο πρώτα. Για παράδειγμα, αν ο χρήστης εισαγάγει την ακολουθία χαρακτήρων "1abc\*D2Efg\_#!\*345Higkl\*mno+op\*qr", το πρόγραμμα να εμφανίζει: Between first two stars (letters:4, digits:1, other:3).

```
#include <stdio.h>
int main(void)
{
    int ch, let, num, stars, other;

    let = num = stars = other = 0;
    printf("Enter characters: ");

    while((ch = getchar()) != '\n' && stars < 2)
    {
        if(ch == '*')
            stars++;

        else if(stars == 1)
        {
            if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
                let++;
            else if(ch >= '0' && ch <= '9')
                num++;
            else
                other++;
        }
    }
    if(stars == 2)
        printf("Between first two stars (letters: %d numbers: %d other:
                %d)\n", let, num, other);
    else
        printf("Never found two stars!\n");

    return 0;
}
```

Άσκηση 2

Να γραφεί ένα πρόγραμμα το οποίο να διαβάξει ένα αλφαριθμητικό και, αν τελειώνει σε αα, να εμφανίζει τους χαρακτήρες του με αντίστροφη σειρά, από το τέλος προς την αρχή.

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

int main(void)
{
    char str[100]; /* Δέσμευση αρκετού χώρου για το αλφαριθμητικό */
    int i, len;

    printf("Enter text: ");
    gets(str); /* Αποθηκεύει του χαρακτήρες στη μνήμη, εκεί όπου
                δείχνει ο pointer str */
    len = strlen(str);

    if(len > 1 && str[len-1] == 'a' && str[len-2] == 'a')
    {
        printf("Reversed text: ");
        for(i = len-1; i >= 0; i--)
            printf("%c", str[i]);
    }
    return 0;
}
```

Άσκηση 3

Ποια είναι η έξοδος του παρακάτω προγράμματος;

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char str[] = "test", *p = str;

    *str = str[strlen(str)];

    while(*p)
        printf("%s\n", p++);

    return 0;
}
```

Απάντηση: δεν εμφανίζει τίποτα

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char str[] = "test", *p = str;

    while(*p)
        printf("%s\n", p++);

    return 0;
}
```

Απάντηση: test

Άσκηση 4

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει ένα αλφαριθμητικό μέχρι 100 χαρακτήρες, τον αριθμό των χαρακτήρων που θα διαγραφούν και τη θέση του πρώτου χαρακτήρα που θα διαγραφεί. Με τη διαγραφή, το υπόλοιπο τμήμα του αλφαριθμητικού να μετακινείται αριστερά τόσες θέσεις όσες και οι χαρακτήρες που διαγράφηκαν. Το πρόγραμμα να εμφανίζει το αλφαριθμητικό πριν τερματίσει. Για παράδειγμα, αν το αλφαριθμητικό είναι "test case" και οι αριθμοί είναι 4 και 2, το πρόγραμμα να εμφανίζει "tcase". Το πρόγραμμα να κάνει έλεγχο εγκυρότητας δεδομένων, ώστε να εξασφαλίζεται ότι οι χαρακτήρες που θα διαγραφούν περιέχονται στο αλφαριθμητικό.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(void)
{
    char str[100];
    int i, len, start, del;
    printf("Enter text: ");
    gets(str);
    len = strlen(str);
    do
    {
        printf("Enter number of characters to delete: ");
        scanf("%d", &del);

        printf("Enter initial position: ");
        scanf("%d", &start);
    } while(start+del > len+1);

    for(i = start-1; i < len-del; i++)
        str[i] = str[i+del];
    str[len-del] = '\0';
    printf("%s\n", str);
    return 0;
}
```

Άσκηση 5α

Να γράψετε μια συνάρτηση με δύο ορίσματα εισόδου: α) μια συμβολοσειρά και β) έναν χαρακτήρα. Η συνάρτηση θα απαλείφει όλες τις (πιθανές) εμφανίσεις του χαρακτήρα μέσα στη συμβολοσειρά και θα επιστρέφει μια ακέραια τιμή που αντιστοιχεί στο πλήθος των εμφανίσεων του χαρακτήρα μέσα στη συμβολοσειρά.

```
int removechar(char *str, char ch)
{
    int cnt=0, i=0;
    char *p;

    p = str;

    while (*p != '\0')
    {
        if (*p != ch)
            *(str+i++) = *p; /* B' τροπος:{ str[i] = *p; i++; } */
            /* C' τροπος:{ *str = *p; str++; } */

        else
            cnt++;
            p++;
    }
    *(str+i) = '\0'; /* B' τροπος: str[i] = '\0'; */
    /* C' τροπος: *str = '\0'; */

    return cnt;
}
```

Άσκηση 5β

Να γράψετε ένα πρόγραμμα που να προτρέπει τον χρήστη να πληκτρολογήσει μια συμβολοσειρά (οι χαρακτήρες μπορεί να είναι αλφαριθμητικοί ή και λευκοί, δηλαδή, κενό ' ' ή tab '\t' αλλά όχι και αλλαγή γραμμής '\n') και να εμφανίζει στην οθόνη τη συμβολοσειρά από την οποία θα έχουν αφαιρεθεί όλες οι εμφανίσεις του χαρακτήρα 'a'. Το πρόγραμμα να χρησιμοποιεί τη συνάρτηση του προηγούμενου ερωτήματος.

```
#include <stdio.h>

int removechar(char *str, char ch);

int main()
{
    int cnt=0;
    char ch, str[256];

    printf("Please enter a sentence:\n");
    gets(str);

    cnt = removechar(str,'a');
    printf("Removed %d characters.\n",cnt);
    printf("The new string is \"%s\".\n",str);

    return 0;
}
```

Άσκηση 6α

Να δημιουργήσετε μια συνάρτηση `cleanstr()` ενός ορίσματος η οποία θα δέχεται μια συμβολοσειρά στην είσοδό της, θα απαλείφει όλους τους αριθμητικούς χαρακτήρες και θα επιστρέφει (με τη λέξη κλειδί `return`) έναν `pointer` στη νέα συμβολοσειρά. Στην περίπτωση που δεν υπάρχουν αριθμητικοί χαρακτήρες, θα επιστρέφει την ίδια συμβολοσειρά.

Π.χ. η `cleanstr("ice10001@uniwa.gr")` θα επιστρέψει δείκτη στη συμβολοσειρά "ice@uniwa.gr" ενώ η `cleanstr("sqrt(a^b)")` θα επιστρέψει "sqrt(a^b)".

```
char *cleanstr(char *instr)
{
    int i=0;
    char *outstr;

    ostr = (char *) malloc(strlen(instr)+1);

    while(*instr != '\0')
    {
        if(*instr < '0' || *instr > '9')
            ostr[i++] = *instr;
        instr++;
    }
    ostr[i] = '\0';
    return ostr;
}
```

Άσκηση 6β

Να γράψετε ένα πρόγραμμα που να προτρέπει τον χρήστη να πληκτρολογήσει 5 συμβολοσειρές και να τις εμφανίζει στην οθόνη απαλείφοντας τους αριθμητικούς χαρακτήρες. Το πρόγραμμα να χρησιμοποιεί την `cleanstr()` του προηγούμενου ερωτήματος.

```
#include <stdio.h>
#include <strings.h>
char *cleanstr(char *);

int main()
{
    int i;
    char line[256];

    for(i=0; i<5; i++)
    {
        printf("Entrer a string: ");
        gets(line);
        printf("Cleaned string: \"%s\\n\", cleanstr(line));
    }

    /* DEN EGINE free h symboloseira pou epistrefei h cleanstr() ! */

    return 0;
}
```

Άσκηση 7

Γράψτε μια συνάρτηση με δύο παραμέτρους, τους ακεραίους `pos` και `cap`. Η συνάρτηση θα επιστρέφει το γράμμα που βρίσκεται στην αντίστοιχη θέση `pos` του λατινικού αλφαβήτου και συγκεκριμένα αν το `cap` είναι 1 θα επιστρέφει το κεφαλαίο ενώ αν είναι 0 θα επιστρέφει το αντίστοιχο πεζό γράμμα. Στην περίπτωση που κάποια παράμετρος είναι εκτός ορίων η συνάρτηση θα επιστρέφει το χαρακτήρα `'\0'`.  
Π.χ.

pos	cap	έξοδος συνάρτησης
4	1	D
4	0	d
26	1	Z
27	0	\0
3	2	\0
-4	1	\0

Απάντηση

```
char gramma(int pos, int cap)
{
    if((pos < 0) || (pos > 26) || ((cap != 0) && (cap != 1)))
        return '\0';

    return ((cap == 1) ? pos+64 : pos+96);
    /* <==> return ((cap == 1) ? pos+'A'-1 : pos+'a'-1); */
}
```