

# Εισαγωγή στον Προγραμματισμό με C, C++

Ασκήσεις Strings

Ιωάννης Τιμπερίδης  
itsimper@cs.duth.gr

# Άσκηση 1

Να γραφτεί πρόγραμμα στη C στο οποίο να διαβάζεται ένα String και να εμφανίζεται ποιος χαρακτήρας είναι αυτός που υπάρχει περισσότερες φορές.

Ένας πιο απλός τρόπος για να διαβαστεί ένα αλφαριθμητικό, σε σχέση με τη χρήση των “scanf” και “fgets”, είναι η Συνάρτηση “**gets**”. Για την ανάγνωση ενός αλφαριθμητικού `str1` εκτελείται το:

```
gets (str1) ;
```

Υπενθυμίζεται ότι στη C ένας χαρακτήρας καταλαμβάνει 1 Byte μνήμης. Δηλαδή, μπορεί να αντιστοιχιστεί στις τιμές από 0 μέχρι και 255.

# Άσκηση 1 (Λύση)

```
char str1[100];
int count[256], i, max;

printf("Which String? "); gets(str1);

for (i=0; i<256; i++) count[i]=0; //Αρχικοποίηση count

i=0; max=0;
while (str1[i] != '\0') { // Μέχρι το τέλος του String
    count[(int)str1[i]]++; // Αύξηση μετρητή
    i++; } // Επόμενος χαρακτήρας

for (i=0; i<256; i++) {
    if (count[i]>count[max]) max = i;}
printf("The most frequent character is %c", max);
```

# Άσκηση 2

Στη C υπάρχει η Συνάρτηση “**strstr**” η οποία αναζητάει εάν υπάρχει ένα αλφαριθμητικό μέσα σε ένα άλλο αλφαριθμητικό. Για παράδειγμα εάν υπάρχει μία λέξη μέσα σε μια φράση. Η Συνάρτηση αυτή όμως επιστρέφει Pointer, που αποτελεί είδος Μεταβλητής που αποθηκεύει θέση μνήμης ως τιμή.

Δεν υπάρχει όμως Συνάρτηση που να εμφανίζει τη θέση μέσα στο αλφαριθμητικό που βρίσκεται το άλλο αλφαριθμητικό.

Να γραφτεί πρόγραμμα στη C που να διαβάζει δύο αλφαριθμητικά και να εμφανίζει τη θέση (δείκτη) του ενός αλφαριθμητικού που ξεκινάει το άλλο αλφαριθμητικό, ενώ να εμφανίζει κατάλληλο μήνυμα εάν δεν υπάρχει.

Programming

Programming

# Άσκηση 2 (Λύση)

```
char frasi[1000],leksi[20];
printf("Which phrase? "); gets(frasi);
printf("Which word? "); gets(leksi);

int i=0,k,found=0;
while (frasi[i]!='\0') { // Μέχρι το τέλος της φράσης
    if (frasi[i]==leksi[0]) {
        found=1; k=1; // Θεωρείται ότι βρέθηκε η λέξη
        while (leksi[k]!='\0') {
            if (leksi[k]!=frasi[i+k]) { // Εάν κάποιος χαρακτήρας...
                found=0; break;} // ...δεν είναι ίδιος
            k++; // Επόμενος χαρακτήρας της λέξης
        }
        if (found) { // Εάν το found παρέμεινε 1
            printf("The word is found at index %d",i); break;
        }
    }
    i++; // Επόμενος χαρακτήρας της φράσης
}
if (!found) printf("The word isn't found");
```

# Άσκηση 3

Στη C δεν υπάρχει Συνάρτηση που να αφαιρεί τα κενά από την αρχή ή από το τέλος ενός αλφαριθμητικού.

Να γραφτεί πρόγραμμα στη C στο οποίο να διαβάζεται ένα αλφαριθμητικό και να διαγράφονται τα κενά από την αρχή και το τέλος του.

Να εμφανίζεται το αλφαριθμητικό πριν και μετά τη διαγραφή των κενών.

Programming

Programming