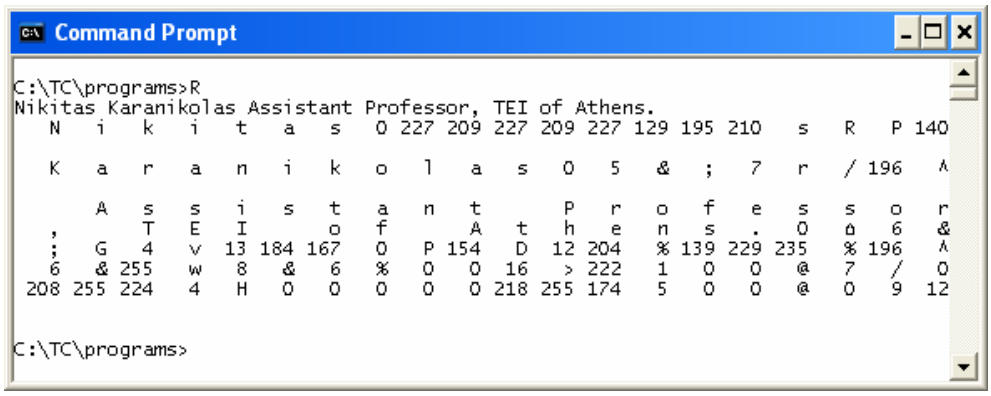
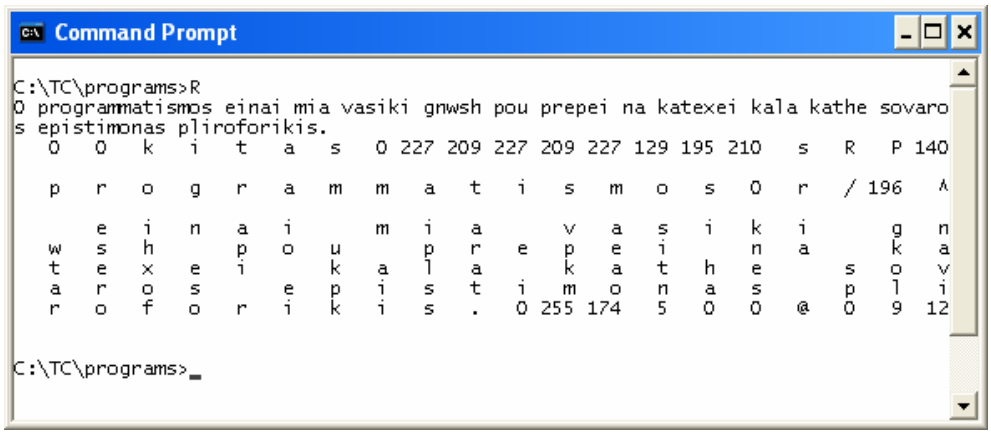


Σε μία άλλη από τις εκτελέσεις του προγράμματος η αλληλεπίδραση με το χειριστή ήταν η επόμενη:



5.5 Συναρτήσεις χειρισμού αλφαριθμητικών

Στις βιβλιοθήκες της γλώσσας C έχουν συμπεριληφθεί μια σειρά από ρουτίνες (συνήθως συναρτήσεις) που απλοποιούν την επεξεργασία των αλφαριθμητικών πινάκων. Παραπάνω αναφέραμε τη ρουτίνα strcpy(). Στη συνέχεια θα περιγράψουμε μερικές ακόμα ρουτίνες χειρισμού αλφαριθμητικών, τις περισσότερο χρήσιμες και διαθέσιμες σε όλες τις εκδόσεις της C.

Πρώτη εξετάζουμε τη συνάρτηση strcmp(), η οποία δέχεται δύο αλφαριθμητικά ορίσματα και επιστρέφει έναν ακέραιο αριθμό. Ο αριθμός αυτός μπορεί να είναι:

- Μηδεν (0) όταν τα δύο αλφαριθμητικά ορίσματα έχουν το ίδιο περιεχόμενο
- Θετικό αριθμό (≥ 1) όταν το πρώτο αλφαριθμητικό όρισμα είναι λεξικογραφικά μεγαλύτερο από το δεύτερο
- Αρνητικό αριθμό (≤ -1) όταν το δεύτερο αλφαριθμητικό όρισμα είναι λεξικογραφικά μεγαλύτερο από το πρώτο

Η αναγκαιότητα της συνάρτησης strcmp() είναι πραγματικά πολύ μεγάλη καθώς η γλώσσα C δεν επιτρέπει την απευθείας (με χρήση των τελεστών ==, !=, <=, >=) σύγκριση αλφαριθμητικών δεδομένων. Η επικεφαλίδα της συνάρτησης είναι:

```
int strcmp(char *s1, char *s2)
```

Αν σας είναι δύσκολο να καταλάβετε την παραπάνω επικεφαλίδα της συνάρτησης strcmp, θεωρείστε ότι αυτή είναι:

```
int strcmp(char s1[], char s2[])
```

Ξαναδιαβάστε την πρώτη από τις δύο επικεφαλίδες της strcmp μετά την ανάγνωση του κεφαλαίου με τους δείκτες.

Το επόμενο πρόγραμμα (έστω y.c) χρησιμοποιείται για να διαβάσει (από το τερματικό) δεδομένα που αφορούν την επιτυχία ή αποτυχία των φοιτητών στην εξέταση κάποιου μαθήματος και να υπολογίσει το ποσοστό επιτυχίας. Η είσοδος των στοιχείων (από το τερματικό) γίνεται με τριάδες αλφαριθμητικών τα οποία αντιπροσωπεύουν το επώνυμο του φοιτητή, το όνομα του φοιτητή και την απόδοσή του στην εξέταση (μια από τις λέξεις «Credit», «Pass» και «Fail»). Το πρόγραμμα διατηρεί τρεις μετρητές (PassCount, NotPassCount, WrongDataCount). Για κάθε γραμμή (τριάδα) δεδομένων αυξάνει τον αντίστοιχο μετρητή. Στο τέλος (όταν ο χρήστης δώσει τη λέξη «QUIT» αντί για όνομα Φοιτητή) το πρόγραμμα υπολογίζει και εμφανίζει

το ποσοστό επιτυχίας καθώς και το πλήθος των λανθασμένων δεδομένων που εισήχθησαν.

```
#include <stdio.h>

main() {
   char surname[30], cname[20], rank[20];
   int PassCount, NotPassCount, WrongDataCount;

   PassCount=NotPassCount=WrongDataCount=0;
   do {
      scanf("%s",surname);
      if ( (strcmp(surname,"quit")==0) ||
           (strcmp(surname,"Quit")==0) ||
           (strcmp(surname,"QUIT")==0) )
         break;
      scanf("%s",cname);
      scanf("%s",rank);
      if ( (strcmp(rank,"Pass")==0) ||
           (strcmp(rank,"pass")==0) ||
           (strcmp(rank,"PASS")==0) ||
           (strcmp(rank,"Credit")==0) ||
           (strcmp(rank,"credit")==0) ||
           (strcmp(rank,"CREDIT")==0) )
         PassCount++;
      else if (
           (strcmp(rank,"Fail")==0) ||
           (strcmp(rank,"fail")==0) ||
           (strcmp(rank,"FAIL")==0) )
         NotPassCount++;
      else
         WrongDataCount++;
   } while (1);
   if ((PassCount>0) || (NotPassCount>0))
      printf("Percentage %d%%\n",
             100 * PassCount / (PassCount + NotPassCount));
   printf("Wrong Data Count: %d%%\n", WrongDataCount);
}
```

Αν στη συνέχεια τρέξουμε αυτό το πρόγραμμα και ως δεδομένα (είσοδο) δώσουμε:

```
Karanikolas Nikitas Pass
Papadakos Panagiotis Credit
```

```
Soulis Stavros Fail
Iordanidis Iordanis Pass
Lalakis Lakis Pass
Zafiriou Manolis FAIL
MITROPOULOS GEORGIOS fail
KARAXRISTOS XRISTOS CREDIT
MOSXOVITIS MARIOS PASS
QUIT
```

Τότε η έξοδος (τα αποτελέσματα) που θα πάρουμε θα είναι:

```
Percentage 66%
Wrong Data Count: 0%
```

Οπως αναφέρθηκε και παραπάνω η εντολή scanf("%s", <variable>) αγνοεί τα κενά και τις αλλαγές γραμμών. Αυτό γίνεται καλύτερα αντιληπτό αν κατά την εκτέλεση του προγράμματος εισάγαγουμε ως δεδομένα:

```
   Karanikolas         Nikitas          Pass
Papadakos                  Panagiotis            Credit
Soulis Stavros Fail
Iordanidis Iordanis                Pass
                   Lalakis Lakis Pass
Vafiadis Sakis Arista
Zafiriou Manolis FAIL
      MITROPOULOS GEORGIOS           fail
KARAXRISTOS             XRISTOS CREDIT
MOSXOVITIS            MARIOS PASS
QUIT
```

Η έξοδος (τα αποτελέσματα) που θα πάρουμε θα είναι:

```
Percentage 66%
Wrong Data Count: 1
```

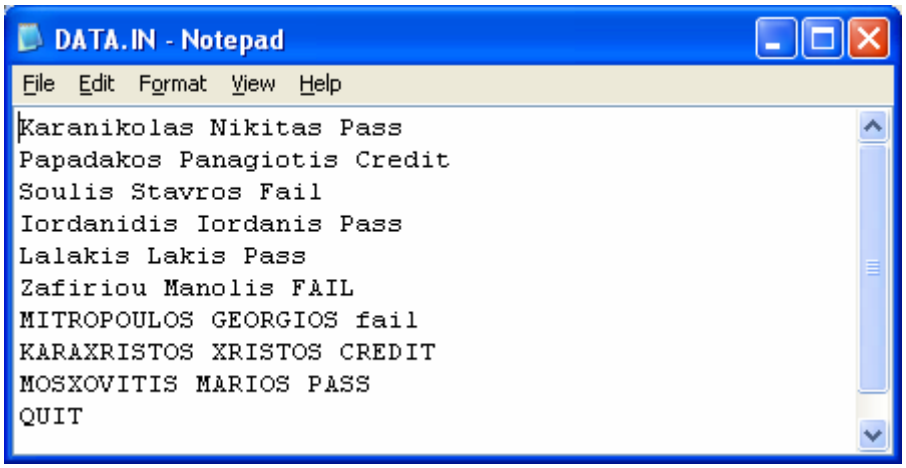
Στην προηγούμενη (δεύτερη) εκτέλεση του προγράμματος προσθέσαμε κενά σε διάφορα σημεία των δεδομένων και μία επιπλέον γραμμή στην οποία βάλαμε την μη κατανοητή (με βάση τη λογική που αναπτύξαμε στο πρόγραμμα) επίδοση «Arista». Στη συνέχεια θα δοκιμάσουμε μία ακόμη (τρίτη) εκτέλεση στην οποία χρησιμοποιούμε τα ίδια δεδομένα με την δεύτερη εκτέλεση αλλά παρεμβάλουμε αλλαγές γραμμών σε διάφορα σημεία της ροής εισόδου. Εστω λοιπόν ότι τα δεδομένα που εισάγουμε είναι:

```
   Karanikolas         Nikitas          Pass
Papadakos                  Panagiotis            Credit
Soulis Stavros

Fail
Iordanidis Iordanis               Pass
                    Lalakis

Lakis Pass
Vafiadis Sakis Arista
Zafiriou Manolis FAIL
      MITROPOULOS GEORGIOS           fail
KARAXRISTOS
             XRISTOS CREDIT
MOSXOVITIS            MARIOS PASS
                    QUIT
```

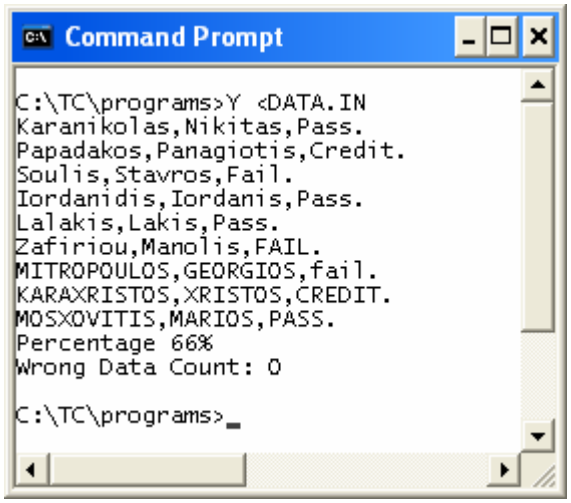
Το αποτέλεσμα από αυτή την (τρίτη) εκτέλεση θα παραμείνει το ίδιο με την δεύτερη εκτέλεση. Αν θέλετε να δείτε εσωτερικά (στις μεταβλητές του προγράμματος) τι γίνεται προσθέστε μια εντολή printf αμέσως πριν το τέλος του do-while-loop. Προσθέστε τη γραμμή:

```
printf("%s,%s,%s.\n",surname,cname,rank);
```

Η προσθήκη της εντολής printf μας δείχνει τα περιεχόμενα των μεταβλητών αλλά αναμειγνύει την είσοδο και την εξοδο του προγράμματος. Αυτό, σε κάποιες περιπτώσεις, μπορεί να είναι ενοχλητικό. Αν θέλετε να αποφύγετε αυτή την ανάμειξη, δακτυλογραφήστε τα δεδομένα σας (ίδια με αυτά που χρησιμοποιήσατε στην τρίτη εκτέλεση) σε έναν εκδότη κειμένου (π.χ. notepad) και αποθηκεύστε τα σε ένα αρχείο (π.χ. data.in). Η επόμενη εικόνα δείχνει το Notepad σε δράση:



Στη συνέχεια εκτελέστε (τέταρτη εκτέλεση) το πρόγραμμά σας ανακατευθύνοντας την είσοδό του ώστε να γίνει από το αρχείο σας (data.in). Η πλήρη αλληλεπίδραση του χρήστη με το λειτουργικό σύστημα (αφού προηγουμένως έχει κατασκευασθεί το αρχείο data.in) είναι:



Θα συνεχίσουμε την παρουσίαση των βασικών συναρτήσεων διαχείρισης String με τη συνάρτηση strcat(). Η συνάρτηση αυτή δέχεται δύο αλφαριθμητικά ορίσματα και προσαρτά (concatenate) στο τέλος του πρώτου τα στοιχεία του δεύτερου string (αλφαριθμητικού). Η συνάρτηση επιστρέφει