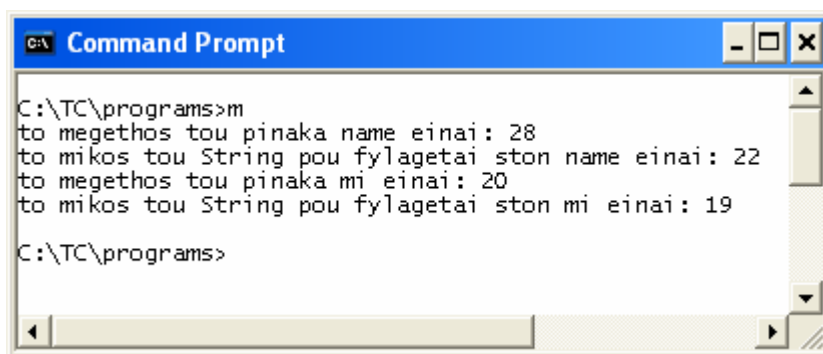


```
strcpy (name, "Nikitas N. Karanikolas");  
printf (  
    "το μέγεθος του πίνακα name είναι: %d\n",  
    sizeof(name));  
printf (  
    "το μήκος του String που φυλάγεται στον name είναι: \  
%d\n", strlen(name));  
printf (  
    "το μέγεθος του πίνακα mi είναι: %d\n", sizeof(mi));  
printf (  
    "το μήκος του String που φυλάγεται στον mi είναι: \  
%d\n", strlen(mi));  
}
```

Η έξοδος θα είναι:



```
C:\TC\programs>m  
το megethos του pinaka name einai: 28  
το mikos του String που fylagetai ston name einai: 22  
το megethos του pinaka mi einai: 20  
το mikos του String που fylagetai ston mi einai: 19  
C:\TC\programs>
```

Μια χρήσιμη λεπτομέρεια που αξιοποιείται από το προηγούμενο πρόγραμμα είναι η χρήση του χαρακτήρα backslash (\), όταν θέλουμε να διακόψουμε ένα αλφαριθμητικό (string) προκειμένου να το συνεχίσουμε σε επόμενη γραμμή του προγράμματος.

5.6 Μελέτη περίπτωσης

Προκειμένου να βελτιώσουμε την κατανόηση των όσων παρουσιάστηκαν, θα κατασκευάσουμε ένα πρόγραμμα που θα καταχωρεί σε έναν πίνακα από αλφαριθμητικά (διδιάστατο πίνακα χαρακτήρων) τα ονοματεπώνυμα των φοιτητών που συμμετείχαν στις εξετάσεις κάποιας εξεταστικής περιόδου και σε ένα δεύτερο πίνακα (μονοδιάστατο πίνακα με ακέραιους) το ποσοστό επιτυχίας του φοιτητή στις εξετάσεις αυτές. Επειδή δεν συμμετέχουν όλοι οι

φοιτητές στον ίδιο αριθμό μαθημάτων, θα πρέπει να επινοήσουμε ένα τρόπο που θα δίνει τη δυνατότητα στο χειριστή, που θα κάνει την καταχώρηση των στοιχείων, να πληροφορήσει το πρόγραμμα ότι ολοκληρώθηκε η καταχώρηση των αποτελεσμάτων για ένα φοιτητή και ότι μπορεί να προχωρήσει στην είσοδο των στοιχείων ενός επόμενου φοιτητή. Η σύμβαση που θα ακολουθούν τα δεδομένα και θα εξυπηρετεί την προαναφερθείσα κατάσταση είναι η εξής:

- Πρώτα εισάγεται το επώνυμο του φοιτητή,
- Κατόπιν εισάγεται το όνομα του φοιτητή,
- Ακολουθούν τα αποτελέσματα του φοιτητή (για κάθε μάθημα που εξετάστηκε αντιστοιχεί μία λέξη από τις «Credit», «Pass» και «Fail»),
- Η ακολουθία αποτελεσμάτων του φοιτητή τερματίζεται με τη λέξη «end».

Η ολοκλήρωση όλων των δεδομένων εισόδου σηματοδοτείται όταν ο χρήστης που κάνει την καταχώρηση δώσει τη λέξη «QUIT» αντί για όνομα Φοιτητή. Για να γίνουν κατανοητά τα παραπάνω ακολουθεί ένα παράδειγμα ροής εισόδου στο πρόγραμμα που επιθυμούμε να κατασκευάσουμε:

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

Όπως προκύπτει και από το παράδειγμα δεδομένων επιτρέπεται η καταχώρηση στοιχείων για κάποιο φοιτητή που δεν εξετάστηκε σε κανένα μάθημα. Προσπαθείστε να κατασκευάσετε το πρόγραμμα μόνοι σας. Αν δεν τα καταφέρετε διαβάστε το πρόγραμμα (έστω w.c) που ακολουθεί:

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

```
main() {
    char names[100][40];
    int percentages[100];
    char surname[30], cname[20], rank[20];
    int PassCount, NotPassCount, WrongCreditCount, percentage;
    int StudentCounter=0;
    int i;

    do {
        scanf("%s", surname);
        if ( (strcmp(surname, "quit")==0) ||
            (strcmp(surname, "Quit")==0) ||
            (strcmp(surname, "QUIT")==0) )
            break;
        scanf("%s", cname);
        PassCount=NotPassCount=WrongCreditCount=0;
        do {
            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 if (
                (strcmp(rank, "End")==0) ||
                (strcmp(rank, "end")==0) ||
                (strcmp(rank, "END")==0) )
                break;
            else
                WrongCreditCount++;
        } while (1);
        if ((PassCount>0) || (NotPassCount>0))
            percentage = 100 *
                PassCount / (PassCount + NotPassCount);
        else
            percentage = 0;
        strcpy(names[StudentCounter], surname);
        strcat(names[StudentCounter], " ");
        strcat(names[StudentCounter], cname);
    }
```

```

percentages[StudentCounter]=percentage;
StudentCounter++;
if (StudentCounter == 100) {
    printf("\nNo more space for data\nThe remaining "
           "data are discarded\n");
    break;
}
} while (1);
for (i=0; i<StudentCounter; i++)
    printf("%s, %d%%.\n",names[i],percentages[i]);
}

```

Στο πρόγραμμα αυτό χρησιμοποιούμε δύο φωλιασμένα `do ... while` επαναληπτικά σχήματα. Το εξωτερικό `do ... while` επαναληπτικό σχήμα είναι υπεύθυνο για τη σάρωση (ανάγνωση και καταχώρηση στους δύο πίνακες) των στοιχείων όλων των φοιτητών. Το εσωτερικό `do ... while` επαναληπτικό σχήμα είναι υπεύθυνο για την ανάγνωση των αποτελεσμάτων του εξεταζόμενου (τρέχοντος) φοιτητή. Η τελευταία δομή ελέγχου `if` μέσα στο εξωτερικό `do ... while` επαναληπτικό σχήμα σκοπό έχει την προστασία του προγράμματος από μεταβολή στοιχείων έξω από τα όρια των δύο πινάκων δεδομένων (`names` και `percentages`). Το επαναληπτικό σχήμα `for` στο τέλος του προγράμματος χρησιμοποιείται για την εμφάνιση των στοιχείων που καταχωρήθηκαν στους πίνακες και εξυπηρετεί στην επαλήθευση της ορθής καταχώρησης των δεδομένων.

Το πρόγραμμά μας είναι αρκετά ικανοποιητικό αλλά αρκετά σπάταλο (σε υπολογιστική ισχύ) γιατί χρησιμοποιεί πολλαπλά `strcmp()` ενώ θα μπορούσε να αντικαταστήσει πολλά από αυτά με ένα. Η επόμενη έκδοση του προγράμματος (έστω `wu.c`) βασίζεται στη συνάρτηση `upper` (που δημιουργούμε εμείς) για να μειώσει τη χρήση της `strcmp()`:

```

#include <stdio.h>

void upper(char *str) {
    int i;
    for (i=0; i<strlen(str); i++)
        if ((str[i] >= 97) && (str[i] <= 122))
            str[i] = str[i] - 32;
}

main() {
    char names[100][40];

```

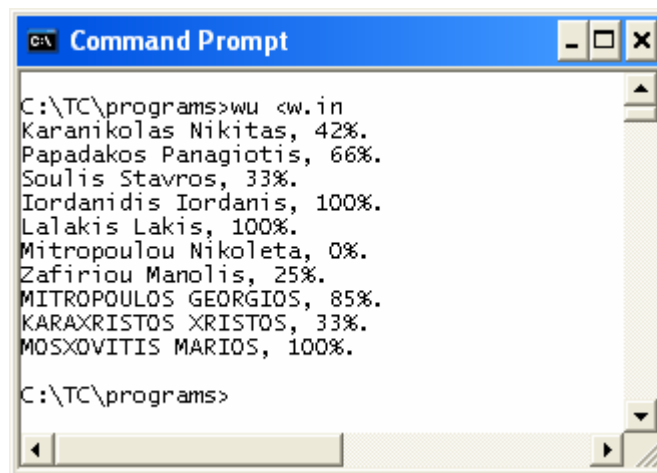
```
int percentages[100];
char surname[30], cname[20], rank[20];
char tmp_surname[30];
int PassCount, NotPassCount, WrongCreditCount, percentage;
int StudentCounter=0;
int i;

do {
    scanf("%s", surname);
    strcpy(tmp_surname, surname);
    upper(tmp_surname);
    if (strcmp(tmp_surname, "QUIT")==0)
        break;
    scanf("%s", cname);
    PassCount=NotPassCount=WrongCreditCount=0;
    do {
        scanf("%s", rank);
        upper(rank);
        if ( (strcmp(rank, "PASS")==0) ||
            (strcmp(rank, "CREDIT")==0) )
            PassCount++;
        else if (strcmp(rank, "FAIL")==0)
            NotPassCount++;
        else if (strcmp(rank, "END")==0)
            break;
        else
            WrongCreditCount++;
    } while (1);
    if ((PassCount>0) || (NotPassCount>0))
        percentage = 100 *
            PassCount / (PassCount + NotPassCount);
    else
        percentage = 0;
    strcpy(names[StudentCounter], surname);
    strcat(names[StudentCounter], " ");
    strcat(names[StudentCounter], cname);
    percentages[StudentCounter]=percentage;
    StudentCounter ++;
    if (StudentCounter == 100) {
        printf("\nNo more space for data\nThe remaining "
            "data are discarded\n");
        break;
    }
} while (1);
for (i=0; i<StudentCounter; i++)
    printf("%s, %d%%.\n", names[i], percentages[i]);
```

```
}

```

Η βασική ιδέα που διέπει τη διαδικασία upper είναι ότι οι τακτικοί αριθμοί των λατινικών κεφαλαίων χαρακτήρων είναι από 65 μέχρι 90 και οι τακτικοί αριθμοί των λατινικών πεζών χαρακτήρων είναι από 97 μέχρι 122. Με βάση αυτή την ιδέα αντικαθιστούμε κάθε πεζό χαρακτήρα με τον αντίστοιχο κεφαλαίο. Για την αντικατάσταση αυτή αρκεί να αφαιρέσουμε τη σταθερά 32 από τον τακτικό αριθμό ενός πεζού λατινικού χαρακτήρα. Το εξελεγμένο πρόγραμμα σε εκτέλεση (wu.exe) καθώς και το αρχείο δεδομένων (w.in) εμφανίζονται στις επόμενες εικόνες:

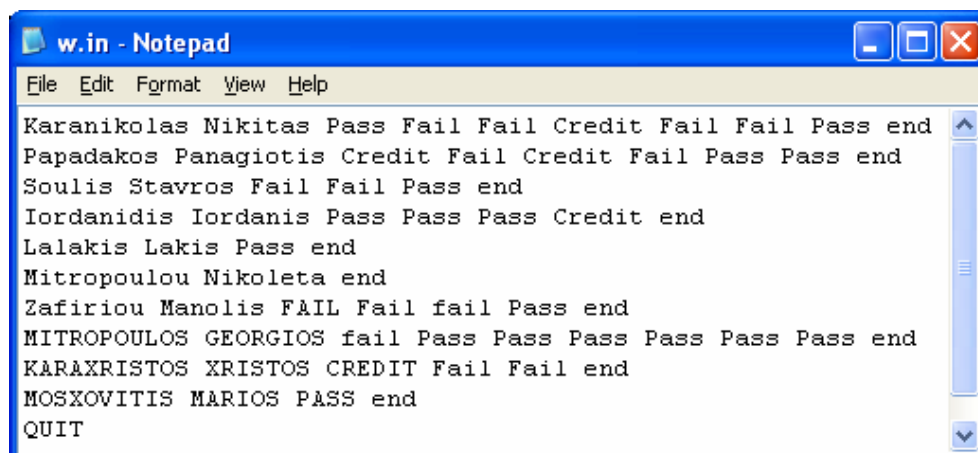


```

C:\TC\programs>wu <w.in
Karanikolas Nikitas, 42%.
Papadakos Panagiotis, 66%.
Soulis Stavros, 33%.
Iordanidis Iordanis, 100%.
Lalakis Lakis, 100%.
Mitropoulou Nikoleta, 0%.
Zafiriou Manolis, 25%.
MITROPOULOS GEORGIOS, 85%.
KARAXRISTOS XRISTOS, 33%.
MOSXOVITIS MARIOS, 100%.

C:\TC\programs>

```



```

File Edit Format View Help
Karanikolas Nikitas Pass Fail Fail Credit Fail Fail Pass end
Papadakos Panagiotis Credit Fail Credit Fail Pass Pass end
Soulis Stavros Fail Fail Pass end
Iordanidis Iordanis Pass Pass Pass Pass Credit end
Lalakis Lakis Pass end
Mitropoulou Nikoleta end
Zafiriou Manolis FAIL Fail fail Pass end
MITROPOULOS GEORGIOS fail Pass Pass Pass Pass Pass Pass end
KARAXRISTOS XRISTOS CREDIT Fail Fail end
MOSXOVITIS MARIOS PASS end
QUIT

```