Fondamenti di Informatica L-A (A.A. 2002/2003) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di Martedì 09/09/2003 – durata 2h:30m

ESERCIZIO 1 (12 punti)

Si scriva una procedura **`action()`** che si occupi di popolare un file di testo chiamato **prestiti.txt (2 punti)**. **`action()`** ha come parametro di ingresso un puntatore a file e deve richiedere all'utente l'inserimento interattivo da tastiera di dati da salvare sul file (puntatore passato come parametro). Il file di testo dovrà contenere richieste di operazioni di prestito/restituzione libri da una biblioteca. Si preveda di terminare l'inserimento dati digitando, ad esempio, '0' come tipo di operazione (vedi seguito). Più precisamente, ogni sua riga include, nell'ordine:

- codice identificativo dell'utente (numero intero), uno e un solo spazio di separazione;
- descrizione sommaria dell'utente (non più di 20 caratteri senza spazi), uno e un solo spazio di separazione;
- tipo operazione (carattere uguale a 'p' per il prestito, 'r' per la restituzione), uno e un solo spazio di separazione;
- codice identificativo del volume di interesse (sempre 6 caratteri), uno e un solo spazio di separazione;
- numero di giorni richiesti/effettivi per il prestito (numero intero sempre positivo).

Ad esempio, **prestiti.txt**:

1772 studente p XX31FD 7
7453 studente r PZ12FV 11
1772 studente p LO66ZX 10
9825 professore p JJ11FF 200
.....

Si scriva una procedura **`load()`** che riceva come parametro di ingresso un puntatore a file di testo (che punterà a **prestiti.txt** di cui sopra) e un intero **`x`**, e restituisca come parametri di uscita un vettore **`y`** contenente strutture **`prestiti`** (codice cliente, codice volume, numero giorni) e il numero degli elementi **`N`** inseriti in **`y`**. **`load()`** deve caricare in **`y`** i prestiti se il valore di **`x`** è negativo, le restituzioni altrimenti **(4 punti)**.

Si scriva un programma C che, utilizzando le procedure **`action()`** e **`load()`** precedentemente definite, inserisca in un vettore **`z`** (supposto di dimensione massima **`DIM=100`**) le sole operazioni di prestito contenute nel file **`prestiti.txt`**. Il programma deve inoltre stampare a terminale tutti i codici relativi ad utenti che hanno più di una operazione di prestito contenuta in **`z`**, purché questa non sia relativa allo stesso volume **(6 punti)**.

ESERCIZIO 2 (6 punti)

Si scriva una funzione **`f()`** che data in ingresso una lista di caratteri ed il carattere **`c`**, restituisca in uscita una nuova lista, ottenuta dalla lista di partenza eliminando gli elementi che precedono eventuali occorrenze di **`c`**. Ad esempio, se invocata con **l=['A','B','C','z','D','x','y','z']** e **c='z'**, la funzione **`f()`** deve restituire la lista **['A','B','z','D','x','z']**.

La funzione **`f()`** può essere realizzata in modo ricorsivo o iterativo, facendo riferimento al tipo di dato astratto **`list`** e alle sue operazioni primitive definite durante il corso, oppure utilizzando direttamente la rappresentazione collegata a puntatori. Si riportino nella soluzione le modifiche da attuare al codice visto a lezione e dovute alla necessità di utilizzare una lista di caratteri e non di interi.

ESERCIZIO 3 (7 punti)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione (si motivi opportunamente la risposta data)?

```
#include <stdio.h>
#include <stdlib.h>
#define L 20
void Proc(char [], int);
int N=L;

main () {
char *s; int i;

s = (char *) malloc(L);
s = "Fondamenti di Infor";
Proc(s,L);
N--; N+2;
printf("Dopo l'invocazione di Proc N vale: %d\n",N);
  {   int i=0;
      for (i=L-2; i>0; i=i-10) printf("%d\n", *(s+i));
  }
}

void  Proc(char y[], int DIM) {
int i, N;
N++; i=1;
printf("Qui N vale: %d\n", N);
while (i<DIM) { y[i]=y[0]; i=i+5; }
```

Esercizio 4 (4 punti)

Si consideri la seguente funzione G:

```
double G(double y){
      if (y>0) {y--; return 2*G(y/3); y++;}
      else return –0.5;
}
```

Si scriva il risultato della funzione quando invocata come **G(7)** e si disegnino i corrispondenti record di attivazione.

Esercizio 5 (2 punti)

Si consideri la grammatica **G** con scopo **S** e simboli terminali **{regina,torre,alfiere,mangia, muove,difende}**

```
S::= A C D | D A | A
D::= A B D | A B
C::= A C C
A::= regina | alfiere
B::= mangia | muove | difende E | difende
E::= torre
```

Si dica se la stringa **regina difende alfiere** è sintatticamente corretta rispetto a tale grammatica e se ne mostri la derivazione *left most*.

Esercizio 6 (2 punti)

Un elaboratore rappresenta i numeri interi su 8 bit dei quali 7 sono dedicati alla rappresentazione del modulo del numero e uno al suo segno. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato, traslandolo poi in decimale per la verifica:

SOLUZIONE

ESERCIZIO 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define DIM 100

typedef struct {int utente; char volume[7]; int giorni;} prestiti;

void action(FILE * F) {
    char tipoOp, descUtente[21], volume[7]; int utente, giorni;
    do {
    printf("Inserire dati da salvare sul file. Per ogni operazione, codice utente,
    descrizione utente, tipo operazione, codice volume, numero giorni.\n Inserire 0
    per terminare");
    scanf("%d %s %c %s %d\n", &utente, descUtente, &tipoOp, volume, &giorni);
    if (tipoOp!='0')
      fprintf(F, "%d %s %c %s %d\n", utente, descUtente, tipoOp, volume, giorni);
    }
    while (tipoOp!='0');
}

void load(FILE * F, int x, prestiti y[], int *N) {
    char tipoOp, descUtente[21], volume[7]; int utente, giorni;
    *N=0;
    while (fscanf(F,"%d %s %c %s %d\n",  &utente, descUtente, &tipoOp, volume,
    &giorni) != EOF) {
      if ((x<0)&&(tipoOp=='p')) {
                y[*N].utente=utente;
                strcpy(volume, y[*N].volume);
                y[*N].giorni=giorni; (*N)++; }
      if ((x>=0)&&(tipoOp=='r')) {
                y[*N].utente=utente;
                strcpy(volume, y[*N].volume);
                y[*N].giorni=giorni; (*N)++; }
      } }

main() {
    prestiti z[DIM]; int N,i,j,count; FILE* f;

    if ((f=fopen("c:\\prestiti.txt", "w"))==NULL) {
      printf("Non sono riuscito ad aprire il file in scrittura!"); exit(1); }
    action(f);
    fclose(f);    //necessario per riportare all'inizio I/O pointer
    f=fopen("c:\\prestiti.txt", "r");

    load(f, -5, z, &N);
    for (i=0; i<N; i++)
      { count=0;
      for (j=i; j<N; j++)
            if ((z[i].utente==z[j].utente) && (i!=j) && strcmp(z[i].volume,
            z[j].volume)) count++;
      if (count>0) printf("Codice cliente con almeno due prestiti di volumi
            differenti: %d\n", z[j].utente);
      }
    fclose(f);
}
```

ESERCIZIO 2

```
typedef struct    list_element {
      char  value; //unico cambiamento da apportare
      struct list_element *next;
} item;
typedef item *list;
list f(list l, char c) {
list prec, 2prec;
if ((l==null)||(l->next==null)) //0 o 1 elemento nella lista
// trovare la piccola imperfezione in questa soluzione…
      return l;
else { 2prec=l; prec=2prec->next; l=prec->next;
      while (l!=NULL) {
            if (l->value==c)
                  { 2prec->next=l; free(prec); }
            l=l->next; }
      } }
```

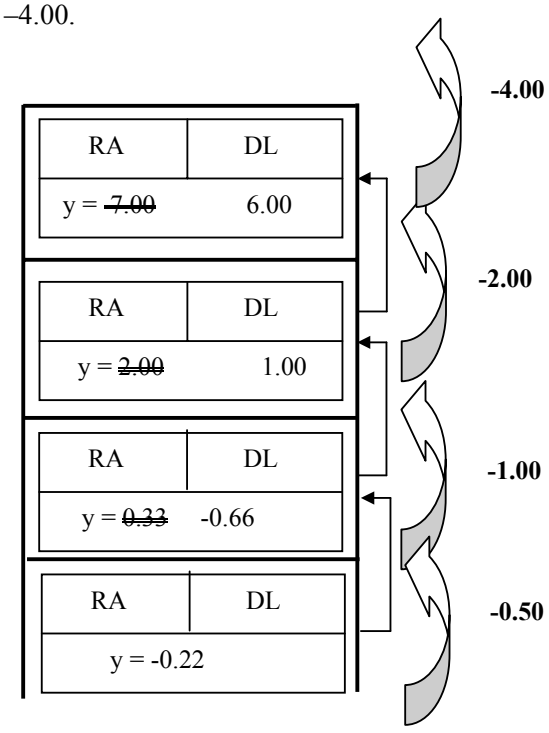
ESERCIZIO 3

Il programma è corretto sintatticamente e stampa:
Qui N vale: 1
Dopo l'invocazione di Proc N vale: 19
114
116

Infatti la stringa s viene inizializzata a "Fondamenti di Infor", terminatore compreso. Successivamente viene invocata la procedura Proc(), alla quale si passa per riferimento la stringa (vettore di char). In particolare, Proc pone uguale a 'F' i caratteri della stringa di posizione 1, 6, 11, e 16. Infine, il programma principale stampa a video i caratteri s[18] e s[8], che non sono stati modificati da Proc() e valgono ancora rispettivamente 'r' e 't'; di tali caratteri viene stampato il codice ASCII corrispondente (formato %d nella printf()).

ESERCIZIO 4

La funzione restituisce il valore –4.00.



ESERCIZIO 5

La stringa è sintatticamente corretta. Derivazione *left most*:

S → D A → A B A → regina B A → regina difende A → regina difende alfiere

ESERCIZIO 6

```
97->       0  1100001
-31->      1  0011111
```

Tra i moduli dei numeri si esegue una sottrazione ottenendo: `0  1000010`
che vale 66 in base dieci.