

Fondamenti di Informatica L-A (A.A. 2002/2003) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di venerdì 19/12/2003 – durata 2h:30m
COMPITO A

ESERCIZIO 1 (12 punti)

Sono dati due file di testo **cineprogramma.txt** e **sale.txt** che contengono, rispettivamente, il programma settimanale dei film in proiezione e le descrizioni delle sale in città. Più precisamente, ogni riga di **cineprogramma.txt** contiene, nell'ordine:

- titolo del film (non più di 30 caratteri senza spazi), uno e un solo spazio di separazione;
- nome della sala (non più di 20 caratteri senza spazi), uno e un solo spazio di separazione;
- 3 orari di inizio proiezione (3 numeri interi separati da caratteri '-'), terminatore di riga.

mentre ogni riga di **sale.txt** contiene, nell'ordine:

- nome della sala (non più di 20 caratteri senza spazi), uno e un solo spazio di separazione;
- costo del biglietto (numero reale), terminatore di riga.

Ad esempio:

cineprogramma.txt	sale.txt
TheKingdom Nosadella 18-20-22	Capitol 6.00
Dogville Fellini 17-20-22	Fellini 5.50
OttoEMezzo Capitol 17-20-23	Modernissimo 6.00
BreakingWaves Odeon 15-19-23	Nosadella 6.50
.....

Si scriva una procedura **load()** che riceva come parametri di ingresso due puntatori a file di testo e restituisca come parametri di uscita un vettore **y** contenente strutture **film** (titolo film, costo biglietto) e il numero degli elementi **N** inseriti in **y**. Per semplicità si supponga che tutte le sale contenute nel primo file siano presenti anche nel secondo, e una sola volta. Si ricorda inoltre l'esistenza della procedura di libreria **void rewind (FILE *f)** che riporta la testina di lettura a inizio file (**6 punti**).

Si scriva un programma C che, utilizzando la procedura **load()** precedentemente definita, inserisca in un vettore **prezzi** (supposto di dimensione massima **DIM=100**) le strutture **film** di cui sopra, derivanti dai file **cineprogramma.txt** e **sale.txt**. Il programma deve inoltre stampare a terminale tutti gli elementi di **prezzi** il cui costo del biglietto è **inferiore alla media** di tutti i costi caricati nel vettore (**6 punti**).

ESERCIZIO 2 (7 punti)

Si scriva una funzione **f()** che date in ingresso due liste ordinate di interi **l1** e **l2**, restituisca in uscita una nuova lista, ottenuta da **l1** eliminando tutti gli elementi contenuti in **l2**. Ad esempio, se invocata con **l1=[-7,-2,0,3,5,8,15]** e **l2=[-1,3,4,5]**, la funzione **f()** deve restituire la lista **[-7,-2,0,8,15]**.

La funzione **f()** può essere realizzata in modo ricorsivo o iterativo, utilizzando il tipo di dato astratto **list** e le operazioni primitive sul tipo **list** definite durante il corso (che quindi possono NON essere riportate nella soluzione).

ESERCIZIO 3 (6 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 12
void Proc(char *, int *);
int N=L;

main () {
char *s; int i;

s = (char *) malloc(L);
s = "Buon Natale";
Proc(s, &N);
N--; N+2;
printf("N vale adesso: %d\n",N);
{int i=0;
for (i=0; i<L; ++i,++i) printf("%c", *(s+i)); }
}

void Proc(char y[], int *DIM) {
int i, N;
N++; i>(*DIM)-4;
while (i>5) { y[i-1]=y[i+1]; i--; }
}
```

Esercizio 4 (6 punti)

Si consideri la seguente funzione **W**:

```
double W(int x){
    if (x<0) {x++; return W(x)+W(x/2); x++;}
    else return -1;
}
```

Si scriva il risultato della funzione quando invocata come **W(-2)** e si disegnino i corrispondenti record di attivazione.

Esercizio 5 (2 punti)

Si consideri la grammatica **G** con scopo **S** e simboli terminali {**fact,exp,(,),1,2,3**}

```
S ::= A C | D B | A
D ::= A B D | A B
C ::= A C C | A B B | BC
A ::= fact | exp | 3
B ::= ( | ) | 3 | 2 | 1
```

Si dica se la stringa **exp (fact 3)** è sintatticamente corretta rispetto a tale grammatica e se ne mostri la derivazione *left most*.

SOLUZIONE COMPITO A

ESERCIZIO 1

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

typedef struct {char titolo[31]; float prezzo;} film;

void load(FILE * f1, FILE * f2, film * y, int * num) {
    char titolo[31], sala1[21], sala2[21];
    int orario; float p;
    *num=0;

    while (fscanf(f1,"%s %s %d-%d-%d\n",titolo,sala1,&orario,&orario,&orario)
        != EOF)
        { while ((fscanf(f2,"%s %f\n",sala2,&p) != EOF)&&
            (strcmp(sala1,sala2))) ; //nessuna azione

            // supponiamo per semplicità che la sala sia sempre compresa in f2
            strcpy(y[*num].titolo, titolo);
            y[*num].prezzo=p;
            (*num)++;
            rewind(f2); }
    }

main() {
    FILE *f1, *f2;
    int N,i; float somma,media;
    film prezzi[DIM];

    if ((f1=fopen("cineprogramma.txt", "r"))==NULL) {
        printf("Non sono riuscito ad aprire file1 in lettura!"); exit(1); }
    if ((f2=fopen("sale.txt", "r"))==NULL) {
        printf("Non sono riuscito ad aprire file2 in lettura!"); exit(1); }

    load(f1,f2,prezzi,&N);
    fclose(f1); fclose(f2);

    for (i=0; i<N; i++) somma=somma+prezzi[i].prezzo;
    media=somma/N;

    for (i=0; i<N; i++)
        if (prezzi[i].prezzo<media)
            printf("Il costo del biglietto per il film %s è %f\n",
                prezzi[i].titolo, prezzi[i].prezzo);
    }
```

ESERCIZIO 2

```
list f(list l1, list l2) {

if (empty(l1) || empty(l2)) return l1;
else {
    if (head(l2)<head(l1)) return f(l1,tail(l2));
```

```

else if (head(l2)==head(l1)) return f(tail(l1),tail(l2));
    else return cons(head(l1), f(tail(l1),l2));
}
}

```

ESERCIZIO 3

Il programma è corretto sintatticamente e stampa:

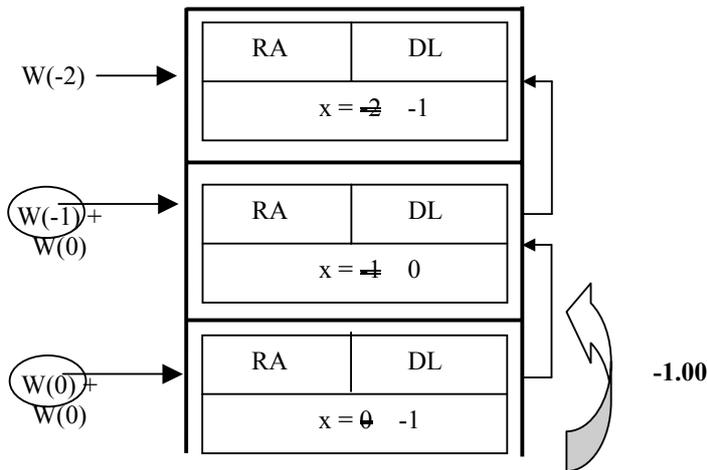
N vale adesso: 11

Bo aae

Infatti la stringa s viene inizializzata a "Buon Natale", terminatore compreso. Successivamente viene invocata la procedura Proc(), di cui è stato dichiarato il prototipo prima del main(), alla quale si passa per riferimento la stringa (vettore di char allocato dinamicamente). In particolare, Proc() assegna ai caratteri di posizione 7,6,5 i valori dei caratteri di posizione, rispettivamente, 9,8,7. Infine, il programma principale stampa a video alcuni caratteri della stringa s, in particolare quelli di posizione pari, a partire da 0 fino a 10, estremi inclusi.

ESERCIZIO 4

La funzione restituisce il valore -3.00. Supponendo che la valutazione degli addendi nella somma venga fatta a partire da sinistra, si ottiene prima:



poi l'eliminazione dell'ultimo record, e un nuovo ulteriore record di attivazione per W(0). Quindi viene fatta la prima somma con risultato -2.00, restituita indietro dove viene fatta l'ultima invocazione di W(0), con somma finale restituita alla prima invocazione di W.

ESERCIZIO 5

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

S → A C → exp C → exp B C → exp (C → exp (A B B →
exp(fact B B → exp(fact 3)