

Fondamenti di Informatica L-A (A.A. 2003/2004) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d’Esame di lunedì 5 aprile 2004 – durata 2h30m
COMPITO B

ESERCIZIO 1 (14 punti)

Un piccolo supermercato ha collegato il proprio registratore di cassa ad un computer, che provvede a registrare in un file binario **scontrini.dat** i dati relativi agli acquisti dei clienti dotati di tessera fedeltà. In particolare nel file binario sono registrate strutture dati **spesa** (si supponga per semplicità che il file contenga al massimo 100 registrazioni), ognuna composta da, nell'ordine:

- nome del cliente (non più di 64 caratteri, senza spazi)
- codice cliente (numero intero)
- nome articolo venduto (non più di 128 caratteri, senza spazi)
- data (numero intero rappresentante il giorno del corrente anno: 1 è il primo gennaio, 365 il trentuno dicembre; l'anno non è bisestile)
- quantità (numero intero)
- importo (numero reale rappresentante l'importo)

Si vuole realizzare un programma di analisi statistica degli acquisti effettuati da uno specifico cliente e analizzare quanto tale cliente ha speso in totale per un determinato prodotto. A tal scopo si scriva una procedura **trova()**, che riceve in ingresso due puntatori a file, **f1** (file binario) ed **f2** (file di testo), un array di caratteri contenente il nome di un cliente e un numero intero passato per riferimento, utilizzato per restituire un risultato.

La procedura deve leggere dal file binario **f1** le strutture di tipo **spesa**, e scrivere sul file di testo **f2** solo alcuni dati relativi ad alcuni acquisti. In particolare si devono considerare solo gli acquisti dei clienti con nome uguale alla stringa passata come parametro. Nel file di testo **f2** devono essere scritti rispettivamente il nome dell'articolo, il giorno di acquisto e l'importo speso, separati da uno spazio; al termine deve essere inserito anche un carattere di "ritorno a capo". La procedura deve infine restituire il numero di record scritti nel file **f2** tramite il quarto parametro passato per riferimento. **(7 punti)**

Si scriva inoltre un programma C che chieda all'utente il nome di un cliente. Il programma deve invocare la procedura **trova()** per scrivere in un file di testo "**stat.txt**" i dati degli acquisti relativi a tale cliente. In seguito, il programma deve chiedere all'utente il nome di un articolo, e stampare a video il nome del cliente, l'articolo scelto, il primo e l'ultimo giorno in cui il cliente ha acquistato tale articolo, e l'importo totale speso dal cliente per quell'articolo. Si consideri a tal scopo di "rileggere" da capo il file **stat.txt**. **(7 punti)**

ESERCIZIO 2 (6 punti)

Si scriva una funzione **diversi()** che date in ingresso due liste **l1** e **l2** di interi, restituisca una terza lista contenente gli interi di entrambe le liste, presi uno per uno alternativamente da **l1** e **l2**. Si devono selezionare solo gli interi che, posti nelle medesime posizioni in entrambe le liste, siano tra loro differenti. Le liste **l1** e **l2** sono non ordinate, ma hanno lunghezza uguale. Ad esempio, se **l1 = [1, 3, 4, 6, 7]** e **l2 = [2, 4, 4, 6, 8]**, il risultato deve essere la lista **[1, 2, 3, 4, 7, 8]**.

La funzione **diversi()** 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>

int modifica(const char[], char*, int*);
int N = 13;

int main () {
    char s1[] = "Bart Simpson", *s2;
    int i;

    s2 = malloc(N);
    for (i=0; i<4; i++)
        s2[i] = s1[i];
    N = modifica(s1, s2, &i);

    printf("i vale adesso: %d\n",i);

    for (i=0; i<N-1; ++i)
        printf("%c", s2[i]);
    return 0;
}

int modifica(const char v1[], char v2[], int *m) {
    int N = 13;
    for (*m=*m+1; *m < N; (*m)++)
        v2[*m] = v1[*m];
    return (*m)++;
}
```

ESERCIZIO 4 (5 punti)

Si consideri la seguente funzione F():

```
int F(float x){
    if ((x == 0) || (x == -1)) return x;
    else return F(--x) - F(--x) - 1; }
```

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

ESERCIZIO 5 (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 COMPITO B

ESERCIZIO 1

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

typedef struct {
    char nome_cliente[65];
    int codice;
    char articolo[129];
    int data; int quantita; float importo;
} spesa;

void trova(FILE *f1, FILE *f2, const char nome[], int *rec) {
    int i, letti = 0;
    spesa tot[DIM];

    *rec = 0;
    letti = fread(tot, sizeof(spesa), DIM, f1);
    for (i=0; i<letti; i++) {
        if (strcmp(tot[i].nome_cliente, nome) == 0) { *rec++;
            fprintf(f2, "%s %d %f\n", tot[i].articolo, tot[i].data, tot[i].importo); }
    }
}

int main() {
    FILE *f1, *f2;
    char art[129], art_temp[129], nome_cliente[65];
    int letti = 0, first = 365, last = 1, num1;
    float tot_importo=0.0, num2;

    if ((f1=fopen("scontrini.dat", "rb"))==NULL) {
        printf("Non sono riuscito ad aprire file1 in lettura!");
        exit(1); }
    if ((f2=fopen("stat.txt", "w"))==NULL) {
        printf("Non sono riuscito ad aprire file2 in scrittura!");
        exit(2); }

    printf("Inserire nome cliente (max. 64 caratteri): ");
    scanf("%s", nome_cliente);
    trova(f1, f2, nome_cliente, &letti);
    fclose(f1); fclose(f2);

    if ((f2=fopen("stat.txt", "r"))==NULL) {
        printf("Non sono riuscito ad aprire file2 in lettura!");
        exit(3); }
    printf("Inserire nome articolo (max. 128 caratteri): ");
    scanf("%s", art);
    while ( fscanf(f2, "%s %d %f\n", art_temp, &num1, &num2) != EOF ) {
        if (strcmp(art, art_temp) == 0) {
            tot_importo = tot_importo + num2;
            if (num1 < first) first = num1;
            if (num1 > last) last = num1; }
    }
    fclose(f2);
    printf("Il cliente %s, dal giorno %d al giorno %d ha speso %.2f euro in %s.\n",
        nome_cliente, first, last, tot_importo, art);
    return 0; }
```

ESERCIZIO 2

```
list diversi(list l1, list l2) {
    if (empty(l1)) return emptyList();
    else if (head(l1)==head(l2)) return diversi(tail(l1), tail(l2));
    else return cons(head(l1),
        cons(head(l2), diversi(tail(l1), tail(l2))));
}
```

ESERCIZIO 3

Il programma è corretto sintatticamente e stampa:

i vale adesso: 14

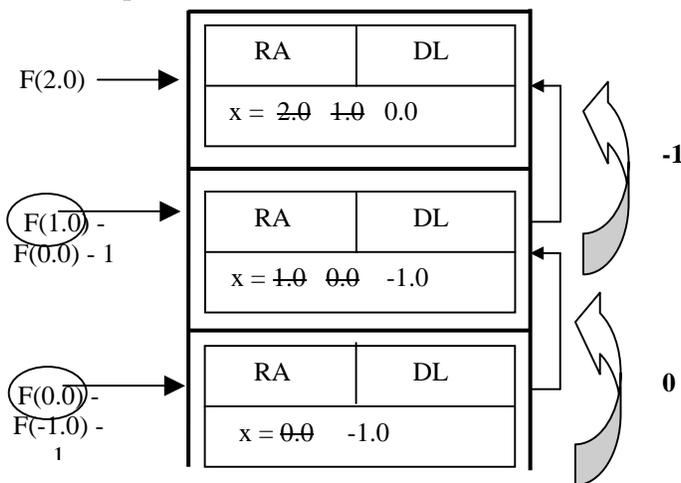
Bart Simpson

La stringa s1 viene inizializzata a "Bart Simpson", terminatore compreso. Successivamente viene allocato dinamicamente uno spazio di N byte, dove N è definito globalmente e vale 13. Un ciclo for si occupa poi di copiare i primi 4 caratteri di s1 su s2, che quindi contiene la scritta "Bart" senza il terminatore di stringa (al momento stringa "mal-formata"). La funzione modifica() viene invocata con parametri s1, s2 e indirizzo di i. Essa copia i rimanenti caratteri di s1 in s2, incrementando direttamente il parametro i passato per riferimento (quindi ci sono dei side-effect). Al termine la funzione restituisce il valore raggiunto da i, che indica quindi anche la dimensione di entrambi gli array, cioè 13. Da notare che nell'istruzione di return c'è un operatore di post-incremento, ed in virtù del passaggio per riferimento, questo opera una modifica sulla variabile i. La funzione restituisce 13, ma i viene a valere 14.

Il main semplicemente stampa il valore della variabile i, che per quanto detto vale 14, e quindi il contenuto dell'array s2, che ora è identico al contenuto di s1.

ESERCIZIO 4

La funzione restituisce il valore -1. 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 F(-1.0). Quindi viene fatta la prima sottrazione con risultato 0, restituita indietro dove viene fatta l'ultima invocazione di F(0.0), con sottrazione finale restituita alla prima invocazione di F().

ESERCIZIO 5

38 -> 0 0100110

114 -> 0 1110010

Tra i moduli dei numeri si esegue una sottrazione ottenendo: 1 1001100
che vale -76 in base dieci.