

Fondamenti di Informatica L-A (A.A. 2004/2005) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di Martedì 19 Aprile 2005 – durata 2h30m
COMPITO B

ESERCIZIO 1 (12 punti)

Una piccola industria di manufatti artigianali utilizza un sistema informatico di controllo qualità sui prodotti. Per ogni manufatto, il sistema registra, su un file binario di nome “**quality.dat**”, strutture dati denominate **quality**, contenenti:

- una stringa **opName**, contenente il nome dell'operatore (al più 64 caratteri, senza spazi);
- un intero **serialNumber**, recante un codice seriale identificativo del manufatto;
- un float **feedback**, contenente un giudizio sulla qualità del manufatto.

Al fine di poter analizzare la qualità dei prodotti in relazione agli operatori, i dirigenti di questa piccola azienda devono poter accedere ai giudizi di qualità. Si deve quindi realizzare un programma che, richiesti il nome dell'operatore e un valore di soglia, scriva su un file di testo i numeri seriali dei prodotti il cui giudizio è inferiore alla soglia. Il nome del file di testo deve essere il nome dell'operatore con l'aggiunta dell'estensione “.err”. Ad esempio, se l'operatore specificato si chiama “Federico”, allora il file dovrà chiamarsi “Federico.err”. Si realizzi:

1. una funzione **filter(...)** che, ricevuti in ingresso un puntatore **source** al file binario, un puntatore **dest** al file di testo, un puntatore a carattere **opName** e un float **threshold**, copi su **dest** i numeri seriali dei prodotti presenti in **source**, relativi all'operatore specificato e con giudizio **feedback** inferiore alla soglia **threshold**. La funzione deve restituire il numero di elementi scritti nel file di destinazione. I numeri seriali devono essere scritti su una sola linea, separati da una virgola, con al termine un carattere di “newline” (“\n”). Al fine di confrontare due stringhe, si utilizzi la funzione **strcmp(...)**, che restituisce 0 se le due stringhe passate come parametri sono identiche. **(punti 6)**
2. un programma C che chieda inizialmente all'utente il nome di un operatore e un valore di soglia (float). Per creare un opportuno nome per il file di destinazione, il candidato può utilizzare le funzioni di libreria:
 - **strcpy(char *s, char *ct)**, che provvede a copiare la stringa **ct** nella stringa **s**;
 - **strcat(char *s, char *ct)**, che concatena il contenuto della stringa **ct** in fondo alla stringa **s** (si faccia particolare attenzione a dimensionare opportunamente la stringa **s** per contenere i 4 caratteri dell'estensione “.err”). La stringa **s**, al termine dell'invocazione, è sempre una stringa ben formata.
 - Dopo aver aperto i file nell'opportuna modalità di lettura/scrittura, il programma utilizzi la funzione **filter(...)** definita al punto precedente per filtrare i dati, e stampi a video quanti elementi sono stati selezionati dalla funzione **filter(...)**. **(punti 6)**

ESERCIZIO 2 (12 punti)

È dato un file di testo, di nome “**data.txt**”, contenente una sequenza di **int** positivi separati da spazi. Non è noto a priori quanti numeri vi siano nella sequenza, e la sequenza può essere terminata dal numero 0 (in tal caso non si deve considerare ciò che è posto dopo il numero 0) o semplicemente dalla fine del file. L'obiettivo è salvare tali numeri in un **array** allocato dinamicamente, la cui dimensione corrisponda esattamente alla lunghezza della sequenza memorizzata sul file. A tale scopo:

1. si definisca una funzione **extract(...)** che, ricevuto un puntatore al file e un intero passato per riferimento, restituisca una lista contenente i valori presenti sul file (si supponga di avere a disposizione le operazioni primitive per la gestione di liste presentate a lezione). La funzione deve inoltre memorizzare

- nell'intero ricevuto come parametro (passato per riferimento) il numero di elementi letti. **(4 punti)**
- si definisca una funzione **invert (...)** che, ricevuta in ingresso una lista **l**, un puntatore **ar** a una zona di memoria già allocata dinamicamente dell'opportuna dimensione, e un intero **size** rappresentante la lunghezza della lista, copi i valori della lista nella zona di memoria, in ordine inverso. **(5 punti)**
 - si definisca un programma **main** che legga dal file "**data.txt**" i valori numerici tramite la funzione **extract (...)**; il programma deve poi allocare dinamicamente memoria sufficiente per contenere i valori letti, e tramite la funzione **invert (...)** deve copiare i valori della lista restituita da **extract (...)** nell'area di memoria allocata dinamicamente. Infine, il programma **main** deve stampare a video i valori caricati nel vettore, utilizzando la notazione sintattica tipica degli array. **(3 punti)**
- Il candidato ipotizzi di avere a disposizione le operazioni primitive sulle liste viste a lezione, che pertanto possono non essere riportate nello svolgimento del compito.

ESERCIZIO 3 (4 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 DIM 12

void scambia(char a[], char *b, int *dim) {
    int i, size = 0;

    *dim = 0;
    for (size=0; a[size] != '\0'; size++);
    for (i=0; i<size-1; ) {
        *(b+i) = a[i+1];
        b[i+1] = *(a+i);
        (*dim)++;
        i=i+2; }
    *(b+size) = '\0';
}

int main () {
    char mese[] = "Aprile";
    char *other;
    int *value, i;

    other = (char *) malloc(DIM * sizeof(char));
    value = (int *) malloc(sizeof(int));
    *value = 0;
    scambia(mese, other, value);
    for (i=0; i < (*value)*2; i++)
        printf("%c", other[i]);
    printf("\n%d\n", *value);
    return 0; }
```

ESERCIZIO 4 (4 punti)

Si descriva brevemente in che cosa consiste la differenza tra una funzione ricorsiva e una funzione ricorsiva tail. Si presentino poi due brevi e semplici esempi di funzione al fine di mostrare tale differenza.

SOLUZIONE

ESERCIZIO 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define DIM 65

typedef struct {
    char opName[DIM];
    int serialNumber;
    float feedback;
} quality;

int filter(FILE *source, FILE *dest, char *opName, float threshold) {
    quality temp;
    int total = 0;

    while (fread(&temp, sizeof(quality), 1, source) > 0) {
        if ((strcmp(opName, temp.opName) == 0) && (temp.feedback < threshold)){
            fprintf(dest, "%d,", temp.serialNumber);
            total++;
        }
    }
    fprintf(dest, "\n");
    return total;
}

int main() {
    char opName[DIM], filename[DIM+4];
    FILE *source, *dest;
    int result = 0;
    float threshold;

    printf("Insert operator's name: ");
    scanf("%s", opName);

    printf("Insert threshold: ");
    scanf("%f", &threshold);

    if ((source = fopen("quality.dat", "rb")) == NULL) {
        printf ("Error opening the file %s\n", "quality.dat");
        exit(-1);
    }
    strcpy(filename, opName);
    strcat(filename, ".err");
    if ((dest = fopen(filename, "w")) == NULL) {
        printf ("Error opening the file %s\n", filename);
        exit(-1);
    }

    result = filter(source, dest, opName, threshold);

    fclose(source);
    fclose(dest);

    printf("Written %d serial numbers.\n", result);
    return 0;
}
```

ESERCIZIO 2

```
#include <stdio.h>
#include <stdlib.h>
#include "list.h"

list extract(FILE *f, int *size) {
    list result;
    int temp;

    result = emptyList();
    *size = 0;
    while ((fscanf(f, "%d", &temp) != EOF) && (temp > 0)) {
        result = cons(temp, result);
        *size = *size + 1; }
    return result;
}

void invert(list l, int * array, int size) { //nota: nessun controllo sulla
    if (empty(l)) // correttezza dei parametri...
        return;
    else {
        array[size-1] = head(l);
        invert(tail(l), array, size-1);
        return; }
}

int main() {
    list l;
    int i, size = 0;
    int * array;
    FILE * f;

    if ( (f = fopen("data.txt", "r")) == NULL ) {
        printf ("Error opening the file %s\n", "data.txt");
        exit(-1); }
    l = extract(f, &size); fclose(f);
    array = (int*) malloc(sizeof(int) * size);
    invert(l, array, size);
    for (i=0; i<size; i++) printf("%d\n", array[i]);
    return 0; }
```

ESERCIZIO 3

Il programma è corretto sintatticamente, viene compilato, ed in esecuzione stampa:

pAirel

3

Nella fase iniziale del programma **main** vengono dichiarate alcune variabili e allocata memoria dinamicamente; in questa fase si inizializza anche a 0 la variabile riferita dal puntatore **value**. Quindi viene invocata la funzione **scambia(...)**: tale funzione esegue inizialmente un ciclo con lo scopo di determinare la lunghezza della stringa **a**, e poi copia, due caratteri alla volta invertendoli, il contenuto di **a** in **b**. La funzione termina copiando anche il terminatore di stringa (al termine della stringa), e restituisce tramite il parametro **dim** il numero di coppie di caratteri copiate, cioè 3.

Il **main** stampa tutti i caratteri della stringa **other**, terminatore escluso, e di seguito il valore puntato da **value** che, per quanto detto, è 3.