

Fondamenti di Informatica L-A (A.A. 2004/2005) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d’Esame di Martedì 21 Giugno 2005 – durata 2h30m

ESERCIZIO 1 (10 punti)

Un server Web tiene traccia degli accessi alle pagine per motivi di sicurezza. Per ogni pagina acceduta, il server scrive su un file binario una struttura dati contenente l’istante di tempo **timestamp** in cui è stata richiesta la pagina (un numero intero), l’indirizzo **ip** del client (una stringa di 15 caratteri), e la dimensione **numBytes** della pagina richiesta in byte (un numero intero). Il numero di elementi registrati nel file non è noto a priori. L’amministratore di sistema ha la necessità di analizzare gli accessi effettuati: a tal scopo si progetti un programma che stampi a video gli istanti di accesso in base a determinati criteri. In particolare, si realizzi:

1. una funzione **readFromFile(...)** che, ricevuti in ingresso un puntatore a un file binario di log del server e un valore intero **atLeast**, restituisca in uscita una lista contenente tutti gli istanti di accesso relativi a file la cui dimensione **numBytes** è maggiore o uguale al parametro **atLeast**. La lista restituita deve essere ordinata in senso crescente. (5 punti)
2. Un programma **main** che, dopo aver richiesto all’utente di inserire il nome del file e la dimensione minima, memorizzi in una lista tutti i **timestamp** relativi a file grandi almeno quanto indicato (utilizzando la funzione **readFromFile**). Dopo aver stampato a video tale lista, il programma chieda all’utente di specificare un intervallo temporale, e stampi a video i valori di **timestamp** contenuti nell’intervallo specificato. (5 punti)

Al fine di svolgere l’esercizio, il candidato consideri di avere a disposizione il tipo di dato astratto **lista**, e le funzioni relative (sia primitive che di alto livello), che pertanto possono non essere riportate nella soluzione.

ESERCIZIO 2 (12 punti)

È dato un file di testo, di nome “**values.txt**”, contenente una sequenza di **int** separati da spazi o da caratteri di **newline**. In particolare, i primi due interi indicano rispettivamente il numero di elementi della sequenza e il valore massimo contenuto nella sequenza; a seguire vi sono poi i numeri interi facenti parte della sequenza. Quindi, se i primi due valori sono detti **size** e **limit**, a seguire nel file ci sono **size** interi memorizzati consecutivamente, il cui valore è compreso tra 0 e **limit** (estremi inclusi, anche valori ripetuti). Si deve costruire un programma che conti quante volte ogni numero intero compare nella sequenza, cioè la frequenza di ogni numero compreso nell’intervallo **[0, limit]**. A tal scopo si realizzi:

1. una funzione

```
int * read(FILE *f, int *size, int *limit)
```

che legga i primi due valori presenti nel file **f**, allochi dinamicamente memoria a sufficienza, e copi gli interi dal file alla zona di memoria. La funzione deve restituire un puntatore all’area di memoria allocata, e tramite i parametri passati per riferimento il numero di interi memorizzati (**size**) e il massimo valore che questi possono assumere (**limit**). (punti 5)
2. Un programma **main** che, aperto opportunamente il file “**values.txt**”, legga da quest’ultimo i valori presenti su file tramite la funzione **read(...)**. Il programma deve poi allocare dinamicamente memoria per mantenere le frequenze di ogni numero nell’intervallo **[0, limit]** (si consideri che tali frequenze costituiscono un vettore di **limit+1** elementi e che ogni elemento di tale vettore può anche assumere il valore 0) e scrivere nella cella **j**-esima di tale array quante volte il numero **j** compare nella sequenza letta dal file. Il programma stampi infine a video il vettore delle frequenze. (punti 7)

ESERCIZIO 3 (5 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>
#include <string.h>

void filter(char a[], char *b) {
    b[0] = *(a+0);
    if ((a[1] == '\0') || (a[2] == '\0')) {
        *(b+1) = '\0';
        return; }
    else {
        b++;
        filter( a+2, b); }
}

int main(int argc, char * argv[]) {
    char faculty[] = "Informatica";
    char * filtered;
    int i;

    filtered = (char *) malloc(strlen(faculty) * sizeof(char) + 1);
    for (i=0; faculty[i] != '\0'; i++)
        filtered[i] = *(faculty + i);
    filtered[i] = '\0';
    printf("%s\n", filtered);

    filter(faculty, filtered);
    printf("%s\n", filtered);
    return 0; }
```

ESERCIZIO 4 (5 punti)

Data la funzione:

```
int squares(int a, int b){
    if (a-b == 0)
        return b;
    else {
        b-=2;
        return squares(a- (b+2), b);
    }
}
```

Mostrare la sequenza dei record di attivazione in seguito all'invocazione `squares(16,7)`. Che risultato viene restituito? La funzione è tail-recursive?

SOLUZIONE

ESERCIZIO 1

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

#define DIM 256

typedef struct {
    int timestamp;
    char ip[16];
    int numBytes;
} log;

list readFromFile(FILE *source, int atLeast) {
    log temp;
    list result = emptylist();
    while (fread(&temp, sizeof(log), 1, source) > 0)
        if (temp.numBytes >= atLeast)
            result = insord(temp.timestamp, result);
    return result;
}

int main() {
    FILE *source;
    int start = 0, end = 0, atLeast = 0;
    char name[DIM];
    list l1, l2;

    printf("Insert file name: ");
    scanf("%s", name);
    printf("Insert min dimension: ");
    scanf("%d", &atLeast);
    if ((source = fopen(name, "rb")) == NULL) {
        printf("Error opening the file %s\n", name);
        exit(-1); }

    l1 = readFromFile( source, atLeast);
    fclose(source);

    l2 = l1;
    while (! empty(l2)) {
        printf("%d ", head(l2));
        l2 = tail(l2);
    }

    printf("Insert startTime and endTime: ");
    scanf("%d%d", &start, &end);

    while((! empty(l1)) && (head(l1) <= end)) {
        if (head(l1) >= start)
            printf("%d ", head(l1));
        l1 = tail(l1);
    }

    return 0;
}
```

ESERCIZIO 2

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

int * read(FILE * f, int * size, int * limit) {
    int * result, temp, i = 0;

    fscanf(f, "%d%d", size, limit);
    result = (int*) malloc(sizeof(int) * (*size));
    while (fscanf(f, "%d", &temp) != EOF) {
        result[i] = temp;
        i++;
    }
    return result; }

int main()
{
    int size, limit, i;
    int * values, *freq;
    FILE * f;

    if ((f = fopen("values.txt", "r")) == NULL) {
        printf ("Error opening the file %s\n", "values.txt");
        exit(-1);
    }
    values = read(f, &size, &limit);
    fclose(f);

    freq = (int*) malloc (sizeof(int) * (limit+1));
    for (i=0; i<=limit; i++)
        freq[i] = 0;
    for (i=0; i<size; i++)
        freq[values[i]] += 1;
    for (i=0; i<=limit; i++)
        printf("%d ", freq[i]);
    return 0; }
```

ESERCIZIO 3

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

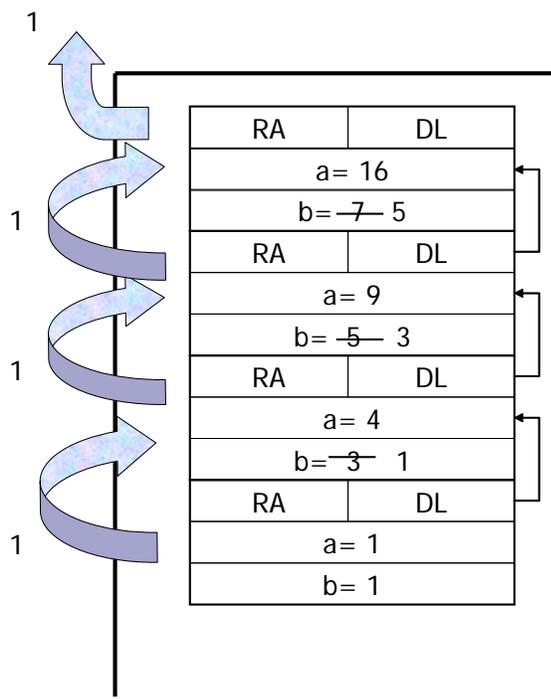
Informatica

Ifraia

Nella fase iniziale del programma **main** vengono dichiarati un array di caratteri (istanziato con la stringa "Informatica"), un puntatore a carattere ed un intero. Il primo gruppo di istruzioni provvede ad allocare memoria sufficiente (riferita tramite il puntatore **filtered**), su cui viene copiata tutta la stringa "Informatica" (è aggiunto anche il terminatore al fine di ottenere una stringa ben formata). Quindi viene stampato il contenuto della memoria puntata da **filtered**, che è appunto "Informatica".

Poi viene invocata la funzione **filter(...)**, che, dopo aver assegnato il primo carattere dell'array a ricevuto come parametro alla prima posizione dell'array **b**, si reinvoca ricorsivamente (incrementando di due posizioni il vettore **a**, e di una posizione il vettore **b**). In pratica la funzione **filter** non fa altro che copiare in **b** tutti i caratteri del vettore **a** che sono in posizioni con indice pari. Il **main** quindi andrà a stampare una stringa contenente solo i caratteri con indice pari della stringa originaria "informatica", cioè "Ifraia".

ESERCIZIO 4



La funzione è ricorsiva tail.