Fondamenti di Informatica L-A (A.A. 2004/2005) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di Venerdì 16 Settembre 2005 – durata 2h30m

ESERCIZIO 1 (10 punti)

Ai fini di effettuare analisi statistiche sull'affluenza agli esami degli studenti di Ingegneria, per ogni esame svolto vengono registrati su un file binario "**`index.dat`**" alcune informazioni. Tali informazioni, organizzate in una struttura apposita, consistono della data d'esame (una stringa di 8 caratteri più un terminatore) e del numero di studenti presenti all'appello (un intero). In particolare, la data è rappresentata tramite 4 cifre per l'anno, due per il mese e due per il giorno (ad esempio il 16 settembre 2005 sarebbe rappresentato con la stringa "**`20050916`**"). Il numero di strutture registrate sul file non è noto a priori. Si vuole realizzare un programma che calcoli l'affluenza media degli appelli precedenti una certa data. A tal scopo il candidato realizzi:

1. una funzione **`readIndex(…)`** che, ricevuti in ingresso il nome di un file e una stringa rappresentante una data, legga in modo opportuno i valori registrati nel file e restituisca una lista di interi. Tale lista dovrà essere composta dai valori di affluenza per gli appelli tenutisi prima della data indicata (escludendo tale data) come parametro. Al fine di confrontare le date, si utilizzi pure la funzione di libreria **`strcmp(…)`**: date consecutive, in virtù della notazione usata, saranno anche consecutive considerando l'ordine lessicografico. (4 punti)
2. un programma **`main`**, che chieda all'utente una data nel formato discusso sopra (si supponga che l'utente inserisca sempre correttamente tale data e si evitino quindi eventuali controlli sul formato, altrimenti necessari). Il programma legga dal file "**`index.dat`**" i valori tramite la funzione **`readIndex`** e poi calcoli la media aritmetica scartando i valori minimo e massimo. A tale scopo il candidato può considerare di creare una seconda lista, contenente i valori della prima in maniera ordinata e quindi semplicemente ignorare il primo e l'ultimo valore. (6 punti)

Al fine di svolgere l'esercizio, il candidato consideri di avere a disposizione il tipo di dato astratto **`lista`** (già definito con elementi di tipo **`int`**), e le funzioni primitive relative, che pertanto possono non essere riportate nella soluzione.

ESERCIZIO 2 (12 punti)

Un programma di statistica degli esami universitari registra su un file "**`indice.txt`**" il numero di studenti promossi ad ogni appello, consecutivamente e separati da uno spazio. In un secondo file, di nome "**`valori.txt`**" sono registrati invece i voti dei promossi per ogni appello; tutti i voti sono registrati consecutivamente, separati solo da uno spazio. Si vuole calcolare la media dei quadrati di tutti i voti. A tal scopo si definisca:

1. una funzione

```
int * readSets(char * fileName, int max, int * dim)
```

   che, noto tramite il parametro **`max`** quanti numeri vi possono essere al massimo nel file **`fileName`**, allochi dinamicamente memoria a sufficienza per un vettore di interi, legga tutti i valori dal file e li salvi nel vettore allocato. La funzione deve restituire un puntatore al vettore e, tramite il parametro **`dim`**, il numero di interi effettivamente letti. (punti 6)

2. Un programma **`main`** che chieda all'utente quanti appelli sono stati registrati al massimo nel file "**`indice.txt`**" e legga il numero dei promossi per ogni appello tramite la funzione **`readSets(…)`**. Fatto ciò, il candidato calcoli quanti voti sono stati salvati nel file "**`valori.txt`**", legga tali valori ancora tramite la funzione **`readSets(…)`**, e infine calcoli e stampi la media dei quadrati dei voti. (punti 6)

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 dim 14

char str[] = "Maggioritario";

void subst(char a, char b, char str[]) {
      int i=0;
      if (*(str+i) != '\0') {
            if (str[i] == a)  str[i] = b;
            subst(a, b, str+1);
      } else return;
}

int main() {
      char * sistema;
      char str[] = "Proporzionale";
      int i;

      subst('o', 'a', str);
      printf("%s\n", str);

      sistema = (char *) malloc(sizeof(char) * 14);
      for (i=dim; i>0; i--) *(sistema+i-1) = str[i-1];
      subst('a', 'o', sistema);
      printf("%s\n", sistema);

      free(sistema);
      return 0;
}
```

ESERCIZIO 4 (4 punti)
Data la funzione:

```
int single_rec(float a, float b){
      if (a/b < 0)
            return b;
      else {
            b= b - (a-b);
            return single_rec(a, b);
      }
}
```

Mostrare la sequenza dei record di attivazione in seguito all'invocazione **`single_rec(5.0, 4)`**. 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"

typedef struct exam {
      char date[9];
      int students;
} exam;

list readIndex(char *fileName, char * date) {
      FILE *findex; exam temp;
      list result = emptylist();

      if ((findex = fopen(fileName, "rb")) == NULL) {
                  printf("Error opening the file %s\n", fileName);
                  exit(-1);}

      while (fread(&temp, sizeof(exam), 1, findex) == 1) {
            if (strcmp(date, temp.date) > 0)
                  result = cons(temp.students, result);
      }
      fclose(findex);
      return result;
}

int main() {
      list participants, partOrdered;
      int i=0, sum=0;
      char date[9];

      printf("Insert date[yyyymmdd]: ");
      scanf("%s", date);

      participants = readIndex("index.dat", date);
      partOrdered = emptylist();
      while (! empty( participants)) {
            partOrdered = insord (head(participants), partOrdered);
            participants = tail(participants);
      }
      partOrdered = tail(partOrdered); // scarto il valore minimo

      while (! empty(tail(partOrdered))) { //scarto il valore massimo
            sum = sum + head(partOrdered);
            i = i + 1;
            partOrdered = tail(partOrdered);
      }

      printf("The average is %f\n", sum/((float) i));
      return 0;
}
```

ESERCIZIO 2

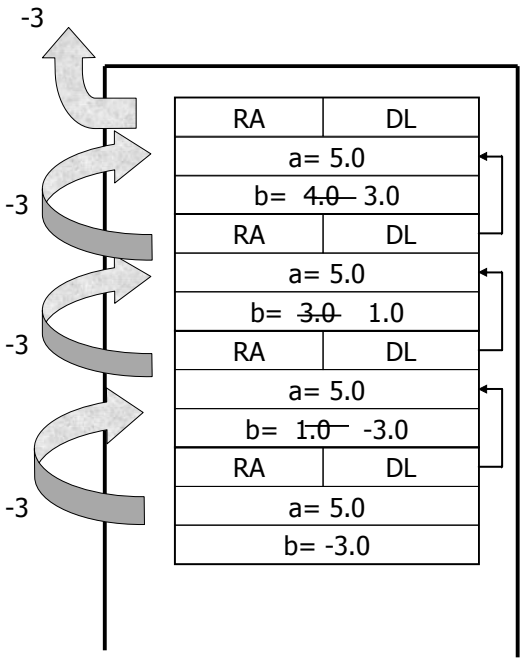
```
#include <stdio.h>
#include <stdlib.h>

int *readSets(char *filename, int max, int *dim) {
      FILE * f;
      int temp; int *buffer;

      if ((f = fopen(filename, "r")) == NULL) {
            printf("Error opening the file %s\n", filename);
            exit(-1); }

      buffer = (int *) malloc(sizeof(int) * (max));
      *dim = 0;
      while(fscanf(f, "%d", &temp) != EOF) {
            buffer[(*dim)] = temp;
            *dim = *dim + 1; }
      fclose(f);
      return buffer; }

int main()
{
      int dimBuffer, i, total = 0, sum = 0, max;
      int *buffer, *valueBuffer;
      FILE * f;

      printf("Numero massimo di valori: ");
      scanf("%d", &max);

      buffer = readSets("indice.txt", max, &dimBuffer);
      for (i=0; i<dimBuffer; i++)
            total = total + buffer[i];

      valueBuffer = readSets("valori.txt", total, &dimBuffer);
      for (i=0; i<total; i++)
            sum = sum + valueBuffer[i] * valueBuffer[i];
      printf("Media dei quadrati: %f", sum / ((float) total)  );

      free(buffer); free(valueBuffer);
      return 0;
}
```

ESERCIZIO 3

Il programma è corretto sintatticamente e la sua esecuzione produce la stampa:
Praparzianale
Proporzionole

La funzione ricorsiva **`subst()`** non fa altro che sostituire nella stringa **`str`** passata come parametro (per riferimento vista l'implementazione tramite **`char *`**), ogni occorrenza del carattere passato con il parametro **`a`** con il carattere passato tramite il parametro **`b`**.
Inizialmente nel **`main`** viene invocata la funzione **`subst()`**, che quindi per come è invocata, sostituisce ad ogni '**`o`**' il carattere '**`a`**'. Quindi viene allocata memoria per **`dim`** (cioè 14) caratteri, e la stringa contenuta in **`str`** viene copiata in tale locazione. La funzione **`subst()`** viene applicata anche a questa seconda stringa, sostituendo questa volta ogni occorrenza del carattere '**`o`**' col carattere '**`a`**'.

ESERCIZIO 4



La funzione è ricorsiva tail.