**Fondamenti di Informatica L-A (A.A. 2004/2005) - Ingegneria Informatica**
**Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di venerdì 14 Gennaio 2005 – durata 2h30m**
**COMPITO A**

## ESERCIZIO 1 (14 punti)

Un negozio di noleggio CD registra, tramite un PC collegato al registratore di cassa, i dati relativi al noleggio dei Compact Disc. Per ogni utente che restituisce un disco, su un file di testo di nome "RentedLog.txt" viene scritto su ogni riga, in ordine:

- un intero **`cd_code`**, identificativo univoco di un cd;
- una stringa, contenente il nome del cliente (al più 64 caratteri, senza spazi);
- un intero **`days`**, che indica la durata in giorni del noleggio.

Dopo aver definito opportunamente una struttura **`rent`** per contenere tali informazioni, il candidato realizzi un programma che chieda all'utente il nome di un cliente e il numero massimo di record che si vogliono ottenere, e stampi a video la lista dei CD noleggiati dal cliente, subito seguito dalla durata media di un noleggio per tale cliente. Con maggior dettaglio:

1. Il candidato scriva una funzione **`readRented(…)`** che riceve in ingresso il nome di un file di testo, il nome di un utente, un puntatore a strutture **`rent`** (che punta ad un'area di memoria opportunamente allocata in precedenza) e la dimensione massima di tale area di memoria (in termini di numero di strutture di tipo **`rent`**). La funzione apra il file e salvi in memoria (tramite il puntatore ricevuto come parametro) i record relativi all'utente specificato (per controllare se un record è relativo al cliente specificato, si utilizzi la funzione **`strcmp(…)`**). La funzione restituisca il numero di record effettivamente letti, che deve risultare minore o uguale alla dimensione massima specificata. Qualora si raggiunga la dimensione massima di record letti prima di aver terminato il file, si ignorino i record rimanenti. **(punti 7)**
2. Il candidato realizzi poi un programma C che chieda inizialmente all'utente il nome di un cliente ed il numero massimo di elementi su cui si vuole effettuare la statistica. Dopo aver allocato dinamicamente memoria sufficiente secondo le istruzioni ricevute dall'utente, il programma utilizzi la funzione **`readRented(…)`** per ottenere i dati relativi al determinato cliente. Si stampi a video poi, in ordine, per ogni CD noleggiato, il nome del cliente, il codice del CD e la durata del noleggio. Si stampi infine la durata media del noleggio. **(punti 7)**

## ESERCIZIO 2 (6 punti)

Si scriva una funzione ricorsiva **`crossSelection()`** che, ricevute in ingresso due liste di interi positivi **`l1`** e **`l2`**, restituisca una terza lista (eventualmente non ordinata) contenente gli interi di **`l2`** che sono nelle posizioni indicate dai valori di **`l1`** (si assuma per convenzione che il primo elemento di una lista sia in posizione 1). Ad esempio, date due liste: **`l1=[1,3,4] e l2=[2,4,6,8,10,12]`**, la lista risultante deve contenere gli elementi di **`l2`** che sono in prima, terza e quarta posizione, cioè: **`[2,6,8]`**.
A tal scopo si realizzi una funzione ricorsiva di supporto **`select()`** che, ricevuti in ingresso una lista e un intero positivo rappresentante una posizione, restituisca l'intero della lista posto alla posizione specificata. La funzione deve restituire -1 qualora l'intero passato non corrisponda a nessuna posizione valida (si assuma comunque positivo l'intero passato).
Le funzioni **`crossSelection()`** e **`select()`** devono essere realizzate in modo ricorsivo, utilizzando il tipo di dato astratto **`list`**. Si possono utilizzare le sole operazioni primitive definite durante il corso (che quindi possono NON essere riportate nella soluzione). Non si possono usare altre funzioni di alto livello.

**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>

#define MAX 16
int falso = 0;

int copia(char s1[], char * s2, int * list, int dim) {
      int i, pos = 0;
      for (i=0; ((i<dim) && (s1[i]!='\0')); i++)
            if (list[i] > 0) {
                  *(s2+pos) = *(s1+i);
                  pos++;  }
      return pos;
}

int main () {
      char a[] = "Europa";
      char * b;
      int * c;
      int i;
      b = (char *) malloc(MAX * sizeof(char));
      c = (int  *) malloc(MAX * sizeof(int) );
      for (i=0; i<MAX; i++)  c[i]=falso;

      for (i=0; a[i] != '\0'; i++)
            if (a[i] != 'a')  c[i]++;
      i = copia(a, b, c, MAX);
      for (; i>0; i--)
            printf("%c", b[i-1]);
      printf("\n");    return 0;  }
```

**ESERCIZIO 4 (3 punti)**

Si consideri la seguente funzione:

```
int div(int a, int b) {
      if (a%b == 0) return b;
      else {
            b++;
            return div(a, b);
      }
}
```

Si scriva il risultato della funzione quando invocata come `div(25, 2)` e si disegnino i corrispondenti record di attivazione. La funzione è ricorsiva tail?

**ESERCIZIO 5 (4 punti)**

Si descriva (in linguaggio naturale e sinteticamente) l'algoritmo di ricerca binaria su un vettore di interi ordinato in senso crescente. Che vantaggio ha tale algoritmo rispetto alla ricerca esaustiva?

## SOLUZIONE

**ESERCIZIO 1**

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

#define DIM 65

typedef struct {
      int cd_code;
      char renter[DIM];
      int days;
} rent;

int readRented(char * fileName, char * name, rent * stat, int maxDim) {
      int dim = 0;
      FILE * f;

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

      while (     fscanf(f, "%d %s %d", &(stat[dim].cd_code),
                        stat[dim].renter, &(stat[dim].days)) != EOF
                  && dim<maxDim) {
            if (strcmp(stat[dim].renter, name) == 0)
                  dim = dim + 1;
      }

      fclose(f);
      return dim;
}

int main() {
      char userName[DIM];
      int maxDim = 0;
      int realDim = 0;
      int i;
      float total = 0;
      rent * stat;

      printf("Inserire nome e dimensione massima: ");
      scanf("%s %d", userName, &maxDim);
      stat = (rent *) malloc(sizeof(rent) * maxDim);
      realDim = readRented("RentedLog.txt", userName, stat, maxDim);

      for (i=0; i< realDim; i++) {
            printf("User: %s, CD: %d, Rented for: %d days\n",
                        stat[i].renter, stat[i].cd_code, stat[i].days);
            total = total + stat[i].days;
      }

      printf("\nAverage length of a rent: %6.2f\n\n", total/realDim);

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

**ESERCIZIO 2**

```
element select(list l, int pos) {
      if (empty(l)) return -1;
      else if (pos == 1 && !empty(l)) return head(l);
            else  return select(tail(l), pos-1);
}

list crossSelection(list l1, list l2) {
      if (empty(l1)) return emptylist();
      else
            return cons(select(l2, head(l1)), crossSelection(tail(l1), l2));
}
```

**ESERCIZIO 3**

Il programma è corretto sintatticamente e stampa:
**poruE**

Il programma **`main`**, dopo una fase iniziale in cui vengono dichiarate le variabili e allocata memoria tramite la funzione **`malloc`** per due array (uno di interi e uno di caratteri), inizializza l'array **`c`** di interi al valore 0. Viene effettuato quindi un ciclo **`for`** sull'array di caratteri **`a`**: se alla posizione i-esima il contenuto dell'array **`a`** è un carattere diverso dalla lettera 'a', allora il corrispondente valore intero nell'array **`c`** viene incrementato.
Viene poi invocata la funzione **`copia`**, la quale si limita a copiare nell'array **`s2`** tutti i caratteri di **`s1`** che sono in posizioni tali per cui il valore intero nella medesima posizione dell'array **`list`** è maggiore di 0. In pratica l'array **`list`** funge da selezionatore per indicare quali caratteri devono essere copiati. Siccome l'array contiene il valore 0 solo nella corrispettiva posizione occupata dalla lettera 'a', in **`s2`** viene copiata la stringa "**`Europ`**", senza terminatore. La funzione copia restituisce poi il numero di caratteri copiati.
Infine il **`main`** provvede a stampare, in ordine inverso, il contenuto della stringa **`b`**, che per quanto detto prima, vale appunto "**`Europ`**". Da notare che **`b`** non è una stringa ben formata, in quanto non presenta il terminatore di stringa '\0' come ultimo carattere.

**ESERCIZIO 4**

La funzione è ricorsiva tail e restituisce il valore 5, attraverso la seguente sequenza di record di attivazione: