

**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 B**

**ESERCIZIO 1 (14 punti)**

Una piccola catena di supermercati registra tramite un PC collegato al registratore di cassa i dati relativi agli scontrini dei clienti dotati di carta fedeltà. Per ogni utente che effettua una spesa, su un file di testo di nome “BillsLog.txt” viene scritto su ogni riga, in ordine:

- una stringa, contenente il nome del cliente (al più 128 caratteri, senza spazi);
- un intero **store**, identificativo unico del supermercato;
- un intero **items**, che indica il numero di beni acquistati;
- un float **price**, che indica il prezzo totale della spesa

Dopo aver definito opportunamente una struttura **bill** per contenere tali informazioni, il candidato realizzi un programma che chieda al gestore del negozio il nome di un cliente e il numero massimo di record che si vogliono ottenere, e stampi a video la lista delle spese effettuate dal cliente, subito seguite dal numero massimo di beni comprati con una sola spesa. Con maggior dettaglio:

1. Il candidato scriva una funzione **readBills (...)** che riceve in ingresso il nome di un file di testo, il nome di un cliente, un puntatore a strutture **bill** (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 **bill**). La funzione apra il file e salvi in memoria (tramite il puntatore ricevuto come parametro) i record relativi al cliente specificato (per controllare se un record è relativo al cliente specificato, si ricordi l’esistenza della funzione di libreria **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 e 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 **readBills (...)** per ottenere i dati relativi ad un determinato cliente. Si stampi a video poi, in ordine, per ogni spesa effettuata, il nome del cliente, il codice del negozio, il numero di oggetti acquistati e il prezzo della spesa. Si stampi infine il numero massimo di beni acquistati in una sola spesa dal quel cliente. **(punti 7)**

**ESERCIZIO 2 (6 punti)**

Si scriva una funzione ricorsiva **crossSelection ()** che, ricevute in ingresso due liste di interi (positivi e negativi) **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 anche una funzione iterativa **select ()** che, ricevuto 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.

Le funzioni **crossSelection ()** e **select ()** devono essere realizzate rispettivamente in modo ricorsivo ed iterativo, 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 20
int vero = 1;

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

    for (i=0; a[i] != '\0'; i++)
        if (a[i] == 'E') c[i]--;
    i = seleziona(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, 8)` 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 decrescente. Che vantaggio ha tale algoritmo rispetto alla ricerca esaustiva?

## SOLUZIONE

### ESERCIZIO 1

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

#define DIM 129

typedef struct {
    char client[DIM];
    int store;
    int items;
    float price;
} bill;

int readBills(char * fileName, char * name, bill * 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, "%s %d %d %f", stat[dim].client, &(stat[dim].store),
        &(stat[dim].items), &(stat[dim].price))!=EOF) && (dim<maxDim)) {
        if (strcmp(stat[dim].client, name) == 0)
            dim = dim + 1;
    }

    fclose(f);
    return dim;
}

int main() {
    char clientName[DIM];
    int maxDim = 0;
    int realDim = 0;
    int i;
    int maxItem = 0;
    bill * stat;

    printf("Inserire nome e dimensione massima: ");
    scanf("%s %d", clientName, &maxDim);
    stat = (bill *) malloc(sizeof(bill) * maxDim);
    realDim = readBills("BillsLog.txt", clientName, stat, maxDim);

    for (i=0; i<realDim; i++) {
        printf("Client: %s, Store: %d, Items: %d days, Price: %f euro\n",
            stat[i].client, stat[i].store, stat[i].items, stat[i].price);
        if (stat[i].items > maxItem)
            maxItem = stat[i].items;
    }
    printf("\nMaximum number of items acquired: %d\n\n", maxItem);

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

## ESERCIZIO 2

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

list crossSelection(list l1, list l2) {
    list temp;
    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:

**aporu**

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 1. Viene effettuato quindi un ciclo `for` sull'array di caratteri `a`: se alla posizione `i`-esima il contenuto dell'array `a` è un carattere uguale alla lettera 'E', allora il corrispondente valore intero nell'array `c` viene decrementato e vale 0.

Viene poi invocata la funzione `seleziona`, 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 vero, cioè diverso da 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 corrispondente posizione occupata dalla lettera 'E', in `s2` viene copiata la stringa "uropu", 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 "uropu". 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:

