

**Fondamenti di Informatica L-A (A.A. 2007/2008) - Ingegneria Informatica**  
**Prof.ssa Mello & Prof. Bellavista – Prova d’Esame di Giovedì 4 Aprile 2008 – durata 2h**  
**Compito A**

**ESERCIZIO 1 (10 punti)**

Una banca vuole memorizzare in un array di strutture dati di tipo **titolo** le informazioni relative ad alcuni titoli azionari. In ogni struttura di tipo **titolo** viene memorizzato il codice unico identificativo del titolo azionario (un intero detto **codTitolo**) e il valore stimato di tale titolo azionario (un float detto **valore**). Il problema sorge dal fatto che ogni sera la Borsa comunica alla banca due array, il primo, di interi, contenente i codici dei titoli (ogni codice è presente due sole volte); il secondo, nelle posizioni corrispondenti al primo vettore, contiene i valori dei titoli. Per ogni titolo, infatti, la Borsa comunica il valore di apertura ed il valore di chiusura, da cui due valori differenti per ogni titolo, e la ripetizione del codice di ogni singolo titolo due volte. Si tenga conto inoltre che la Borsa memorizza i codici dei titoli in ordine casuale, e quindi il valore di apertura ed il valore di chiusura dello stesso titolo spesso non sono in posizioni consecutive.

a) Il candidato realizzi una funzione:

```
titolo * valMedio(int codTitoli[], float valori[], int dim, int * resultDim)
```

che riceva come parametri un vettore di interi rappresentate i codici dei titoli **codTitoli**, un vettore di float, detto **valori**, rappresentante i valori di ogni titolo (il valore di un titolo è memorizzato nella posizione corrispondente al codice del titolo nel primo vettore), e la loro dimensione **dim**; questi due array sono i dati che la Borsa ogni sera comunica alla banca. La funzione deve calcolare il valore medio di ogni titolo. A tal scopo, la funzione deve restituire in un apposito vettore di strutture di tipo **titolo**, allocato dinamicamente (non necessariamente della dimensione strettamente necessaria), il codice di un titolo e il suo valore medio (senza titoli ripetuti). Tramite il parametro **resultDim**, passato per riferimento, la funzione restituisce anche la dimensione logica dell’array di strutture **titolo**. Si riporti, per completezza, anche la definizione della struttura dati **titolo**.

b) Il candidato realizzi poi un semplice **main** di esempio, in cui vengono dichiarati e definiti due opportuni array rappresentanti codici di titoli e valori (simulando così i dati provenienti dalla Borsa), e si usi la funzione di cui al punto a) con tali dati ipotetici per generare un vettore di strutture **titolo** e stamparne a video le coppie codice titolo/valore medio.

**VARIANTE SOLO PER VECCHIA MODALITÀ:** In aggiunta a quanto specificato sopra, il candidato realizzi una funzione che, ricevuto come parametro il nome di un file binario, provveda a scrivere su tale file binario le strutture dati di tipo **titolo** determinate al punto b). In particolare, si scriva sul file innanzitutto il numero di strutture che vi verranno memorizzate e poi, a seguire, le strutture dati stesse. Al fine di controllare che la fase di scrittura sia andata a buon fine, si riapra il file e se ne legga il contenuto, confrontando i valori letti da file con quelli in memoria, e segnalando eventuali incongruenze.

**ESERCIZIO 2 (9 punti)**

Si supponga di avere a disposizione, già definiti, l’ADT lista per interi (denominato **list**, con relative primitive). Il candidato definisca una funzione *ricorsiva*

```
list max_l(list l1)
```

che, ricevuta in ingresso una lista **l1** di interi, restituisca una nuova lista contenente solo gli interi della lista di ingresso tali per cui l’elemento successivo è minore o uguale di quello in considerazione, o non esiste. Nel caso in cui l’elemento considerato sia effettivamente maggiore del successivo, tale successivo elemento non deve essere più preso in considerazione, ma semplicemente ignorato. Ad esempio, se invocata con **l1 = [8, 4, 3, 7, 2, 5]**, la funzione deve restituire la lista **[8, 7, 5]**. Infatti **8 > 4**, e quindi **8** viene selezionato e **4** successivamente ignorato; poi viene scartato il **3** perché **3 < 7**, viene selezionato il **7** (**2** ignorato), ed infine viene selezionato il **5** perché ultimo numero.

### **ESERCIZIO 3 (7 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 <string.h>
int dim = 0;

int select(char * s1, char * s2) {
    int result = 0;
    while (*s1!= '\0' && *(s1+1)!='\0') {
        if (*s1>*(s1+1)) {
            if (*(s2)!='\0') {
                *s2 = *s1;
                dim = dim+1;
                s2 = s2 + 1;
            }
        }
        s1 = s1+1;
    }
    return dim;
}

int main(void) {
    int pos;
    char s1[] = "info";
    char * s2 = (char *) malloc(sizeof(char) * 11);
    strcpy(s2, "ingegneria");
    pos = select(s1, s2+2);
    printf("%d\n", pos);
    while (*s2 != '\0') {
        printf("%c", *s2);
        s2++; }
    return (0); }
```

### **ESERCIZIO 4 (3 punti)**

Il candidato descriva brevemente come, nel linguaggio C, avviene il passaggio dei parametri nell'invocazione di una funzione. Il candidato abbia cura di corredare la spiegazione con brevi esempi di codice.

### **ESERCIZIO 5 (3 punti)**

Un elaboratore rappresenta i numeri interi su 8 bit tramite la notazione in complemento a 2. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato traslandolo poi in decimale per la verifica:

$$67+(-94)$$

## ESERCIZIO 1

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

typedef struct {
    int codTitolo;
    float valore;
} titolo;

titolo * valMedio(int codTitoli[], float valori[], int dim, int * resultDim) {
    int i, j;
    titolo * result;
    int presente;

    result = (titolo *) malloc((dim/2) * sizeof(titolo));
    *resultDim = 0;

    for (i=0; i<dim; i++) {
        presente = 0;
        for (j=0; j<*resultDim && !presente; j++)
            if (codTitoli[i] == result[j].codTitolo)
                presente = 1;
        if (!presente) {
            for (j=i+1; j<dim && (codTitoli[i] != codTitoli[j]); j++);
            result[*resultDim].codTitolo = codTitoli[i];
            result[*resultDim].valore = (valori[i] + valori[j])/2;
            *resultDim = *resultDim + 1;
        }
    }
    return result;
}

void scriviBin(char * filename, titolo media[], int dim) {
    int i;
    FILE * fp;
    titolo temp;

    fp = fopen(filename, "wb");
    fwrite(&dim, sizeof(int), 1, fp);
    fwrite(media, sizeof(titolo), dim, fp);
    fclose(fp);

    fp = fopen(filename, "rb");
    fread(&i, sizeof(int), 1, fp);
    if (i!=dim)
        printf("Errore nel numero di items!\n");
    for (i=0; i<dim;i++) {
        fread(&temp, sizeof(titolo), 1, fp);
        if ((temp.codTitolo != media[i].codTitolo) ||
            (temp.valore != media[i].valore))
            printf("Errore al record numero %D\n", i);
    }
    fclose(fp);
}

int main(void) {
    titolo * media;
    int mediaDim;
```

```

int i;
int codTitoli[] = {1,2,3,2,3,1};
float valori[] = {2,4,6,8,12,4};

media = valMedio(codTitoli, valori, 4, &mediaDim);
for (i=0; i< mediaDim; i++)
    printf("%d %f\n", media[i].codTitolo, media[i].valore);
scriviBin("prova.bin", media, mediaDim);
return (0);
}

```

## ESERCIZIO 2

```

list max_1(list l1)
{
    if (empty(l1))
        return emptylist();
    else
    {
        if (empty(tail(l1)))
            return cons(head(l1), emptylist());
        else
        {
            if (head(l1) > head(tail(l1)))
                return cons(head(l1), max_1(tail(tail(l1))));
            else
                return max_1(tail(l1));
        }
    }
}

```

## ESERCIZIO 3

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

```

1
innegneria

```

Nella funzione main() vengono assegnate alle variabili s1 e s2 due stringhe, e poi viene invocata la funzione select(). Questa controlla ogni carattere di s1: se questo è maggiore del successivo allora cerca di copiarlo in s2, previo controllo che s2 attualmente non punti al carattere di terminazione delle stringhe. Nel caso in cui venga effettuata la copia, s2 viene incrementato, così come vien incrementata anche la variabile globale dim.

Nell'invocazione di select(), l'unico carattere il cui valore è maggiore del successivo è 'n'; siccome poi la funzione viene invocata nel main con parametro attuale "gegneria" (nel main la funzione è invocata con s2+2), viene sostituito solo il primo carattere, e al posto di 'g' viene messo 'n'. Al termine della funzione, il main stampa il numero di caratteri sostituiti (1), e poi la stringa s2 iniziale, cioè innegneria.

## ESERCIZIO 5

```

67  -> 01000011          01000011+
94  -> 01011110          10100010=
      10100001          -----
-94 -> 10100010          11100101 ->-27
                        00011010
                        00011011 ->27

```