

Fondamenti di Informatica L-A (A.A. 2007/2008) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di Giovedì 10 Luglio 2008 – durata 2h
Compito A

ESERCIZIO 1 (11 punti)

La Comunità Europea ogni anno valuta le proposte di progetti di ricerca e i relativi finanziamenti. A tal fine per ogni proposta viene compilata un'apposita struttura dati di nome **proposta**, contenente un campo stringa col nome del progetto (di al più 63 caratteri), un intero rappresentante un voto ricevuto dai valutatori, e l'importo finanziario richiesto dal progetto (un float). Dato l'importo totale disponibile e la lista dei progetti, è necessario determinare quali progetti saranno finanziati, sulla base ovviamente del voto ricevuto e della disponibilità economica. Si noti che la disponibilità economica residua va diminuendo mano a mano che si finanziano i progetti dipendentemente dall'importo richiesto dagli stessi: se ad esempio l'importo iniziale disponibile è di Euro 1.000.000 e il primo progetto finanziato (voto massimo dai valutatori) prevede una richiesta di Euro 300.000, per i restanti progetti il finanziamento disponibile rimarrà di Euro 700.000. Inoltre la Comunità Europea, per motivi di correttezza, finanzia solo i primi n progetti il cui costo è effettivamente coperto dal finanziamento disponibile: ad esempio, se dopo aver finanziato i 5 progetti migliori rimangono a disposizione Euro 150.000, ed il sesto progetto prevede un finanziamento di Euro 200.000, questo non viene finanziato; supponendo poi che il settimo miglior progetto richieda un finanziamento di soli Euro 30.000, questo non viene comunque finanziato per correttezza.

a) Dopo aver definito opportunamente la struttura dati **proposta**, il candidato realizzi una funzione:

```
proposta * vincitori(proposta * proposte, int dim, int * dimResult, float money)
```

che riceva come parametri di ingresso un vettore **proposte** di strutture dati di tipo **proposta**, la dimensione di tale vettore **dim** e l'ammontare totale iniziale del finanziamento disponibile, detto **money**; la funzione deve restituire un nuovo vettore allocato dinamicamente, contenente i primi n progetti finanziabili tramite il finanziamento **money**; la somma dei finanziamenti dei primi n progetti finanziabili ovviamente non deve eccedere il finanziamento disponibile. La dimensione logica del vettore risultato deve essere restituita tramite il parametro passato per riferimento **dimResult**. Si noti che il vettore in ingresso **proposte** non è ordinato. Inoltre, al fine di minimizzare le discussioni con i proponenti dei progetti, i valutatori hanno fatto in modo che non vi siano due progetti con lo stesso voto.

b) Il candidato realizzi poi un semplice **main()** di esempio, in cui viene dichiarato e definito un array di strutture dati **proposta**, con valori opportunamente inizializzati e si usi la funzione di cui al punto a) per determinare i progetti finanziati. Si stampi a video l'elenco dei progetti finanziati e il residuo di finanziamento (eventualmente pari a 0).

ESERCIZIO 2 (8 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 somme2(list l1)
```

che, ricevuta in ingresso una lista **l1** di interi, restituisca una nuova lista contenente solo gli interi della lista di ingresso il cui valore è uguale alla somma dei due interi seguenti nella lista (a tal fine, gli ultimi due numeri di una lista sono automaticamente esclusi). Ad esempio, se invocata con **l1 = [5, 6, 4, 2, 1, 1, 3, 1]**, la funzione deve restituire la lista **[6, 2]**. Infatti, considerando il primo valore (5): i due valori successivi sono 6 e 4, e la loro somma vale 10; quindi 5 è scartato. Per il secondo valore, pari a 6, la somma dei due valori successivi è proprio 6 e quindi il valore viene selezionato per la lista risultato.

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 num = 8;

int compare(char * a, char * b, int num) {
    int i = 0;
    while (*(a+i+1)!='\0' && *(a+i) != *(a+i+1)) {
        if (a[i] != b[i])
            b[i] = a[i];
        i++;
    }
    a[i] = b[i] = '\0';
    if (!strcmp(a,b))
        return num-i;
    else
        return num;
}

int main(void)
{
    int result;

    char s1[] = "capperi";
    char s2[8];
    strcpy(s2, "dattero");
    result = compare(s1, s2, num);
    printf("%d\n", result);
    if (num-result >= 0 ) {
        printf("%s\n", &(s1[num-result+1]));
        printf("%s\n", s2+num-result+1);
    }
    return (0);
}
```

ESERCIZIO 4 (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:

$$90+(-109)$$

ESERCIZIO 5 (3 punti)

Data la funzione:

```
int mm(int p, int q) {
    if( p-q > q) return p;
    else return mm(q-2,p-2) + mm(q-1,p-1);
}
```

mostrare la sequenza dei record di attivazione nel caso in cui la funzione sia invocata con parametri attuali (1, 1).

ESERCIZIO 1

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

typedef struct {
    char nome[64];
    int voto;
    float money;
} proposta;

proposta * vincitori2(proposta * proposte, int dim, int * dimResult, float money) {
    proposta * result;
    int * giaFinanziati;
    int i, maxVoto, maxPos;

    *dimResult = 0;
    result = (proposta *) malloc(dim * sizeof(proposta));
    giaFinanziati = (int *) malloc(dim * sizeof(int));
    for (i=0; i<dim; i++)
        giaFinanziati[i]=0;
    do {
        maxVoto = maxPos = -1;
        for (i=0; i<dim; i++) {
            if (proposte[i].voto>maxVoto && giaFinanziati[i]==0) {
                maxPos = i;
                maxVoto = proposte[i].voto; }
        }
        if (maxPos > -1) {
            result[*dimResult] = proposte[maxPos];
            *dimResult = *dimResult + 1;
            money = money - proposte[maxPos].money;
            giaFinanziati[maxPos] = 1; }
    } while (money > 0 && maxPos > -1);
    free(giaFinanziati);
    *dimResult = *dimResult - 1;
    return result; }

proposta * leggiFile(char * filename, int * dim) {
    FILE * fp;
    proposta * p;
    if ((fp=fopen(filename, "r")) == NULL) exit(-1);
    fread(dim, sizeof(int), 1, fp);
    p = (proposta *) malloc(sizeof(proposta) * *dim);
    fread(p, sizeof(proposta), *dim, fp);
    fclose(fp); return p; }

int main(void) {
    proposta * p, * p2;
    int dimResult;
    int max, i;
    float temp;

    p = leggiFile("proposte.dat", &max);
    temp = 30;
    p2 = vincitori(p, max, &dimResult, temp);
    for (i=0; i<dimResult; i++) {
        printf("%s %d %6.2f\n", p2[i].nome, p2[i].voto, p2[i].money);
        temp = temp - p2[i].money; }
    printf("Residuo finanziario: %6.2f\n", temp); return (0);
}
```

ESERCIZIO 2

```
list somme2(list l1) {
  if (empty(l1) || empty(tail(l1)) || empty(tail(tail(l1))))
    return emptylist();
  else {
    if (head(l1) == head(tail(l1)) + head(tail(tail(l1))))
      return cons(head(l1), somme2(tail(l1)));
    else
      return somme2(tail(l1));
  }
}
```

ESERCIZIO 3

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

```
6
peri
tero
```

Nella funzione main() vengono assegnate alle variabili s1 e s2 due stringhe, e poi viene invocata la funzione compare(). Questa funzione scorre il contenuto del primo array, alla ricerca di due caratteri consecutivi identici; finché non li trova, confronta i caratteri delle due stringhe nella stessa posizione: se sono diversi, copia un carattere della prima stringa nella seconda stringa, alla corrispondente posizione. Appena trova due caratteri consecutivi identici, si ferma e pone in tale posizione un terminatore di stringa. Quindi invoca una strcmp sulle due stringhe così modificate, che risultano identiche, e restituisce il valore di num meno la posizione dove ha trovato i due caratteri identici. Nel caso particolare restituisce 8-2, cioè 6.

Il main poi si limita a stampare il contenuto delle stringhe dopo l'occorrenza dei due caratteri uguali, contenuto che non è stato modificato. Quindi viene stampato peri e poi tero.

ESERCIZIO 4

```
90   -> 01011010           01011010+
109  -> 01101101           10010011=
      10010010           -----
-109 -> 10010011           11101101 ->-19
                                00010010
                                00010011 ->19
```

ESERCIZIO 5

