

Fondamenti di Informatica L-A (A.A. 2008/2009) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d’Esame di Lunedì 22 Dicembre 2008 – durata 2h
Compito A
Compito Modalità Fondamenti di Informatica L-A (solo a.a. precedenti)

ESERCIZIO 1 (10 punti)

Un’agenzia turistica vuole offrire ai suoi clienti le proposte sciistiche nelle migliori località possibili. A tal scopo tiene traccia di alcuni dati relativi al manto nevoso, e poi di volta in volta seleziona le località che rispecchiano solo certi criteri. Ogni località è rappresentata tramite una struttura dati di nome `pista`, che contiene il nome della località (al più 31 caratteri), il codice identificativo unico (un intero), un array di interi, di nome `elenco`, rappresentante i valori misurati del manto nevoso (l’array ha dimensione fisica di 120 valori interi indicanti i cm. di neve), ed un ulteriore intero `dim` che specifica quante misurazioni sono disponibili (cioè la dimensione dell’array `elenco`). Purtroppo per alcune località sono disponibili misurazioni giornaliere, mentre di altre località si hanno misurazioni sporadiche: quindi a seconda della località il numero di misurazioni cambia.

Dopo aver definito la struttura dati `pista`, il candidato definisca ed implementi la funzione:

```
int * seleziona(pista p[], int dim, int *dimResult);
```

che, ricevuti in ingresso un array `p` di strutture dati di tipo `pista` e la sua dimensione `dim`, selezioni solo le piste che hanno avuto più di 100 cm. di neve per almeno un terzo delle misurazioni disponibili per quella pista. L’elenco delle località selezionate deve essere restituito tramite un array di interi allocato dinamicamente, dove tali interi sono i codici identificativi delle località stesse. La dimensione dell’array risultato deve essere restituita tramite il parametro `dimResult` passato per riferimento.

Si realizzi poi un `main` che provveda a leggere da un file binario di nome “`valori.dat`” le strutture dati rappresentanti le piste, invochi la funzione `seleziona(...)` e stampi a video i codici delle località restituite dalla funzione. Si noti che nel file sono registrate al massimo 100 strutture dati, ma che il numero esatto non è noto a priori.

ESERCIZIO 2 (10 punti)

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

```
list selectMax(list l1, list l2);
```

Le liste `l1` ed `l2`, della stessa lunghezza, contengono i valori di temperatura letti da due termometri posti in due punti diversi della città. Ogni valore rappresenta una misurazione effettuata ad una certa ora del giorno: il valore in una certa posizione nella lista `l1` è la temperatura letta ad una certa ora nel primo termometro; il valore di `l2` memorizzato nella stessa posizione rappresenta la temperatura del secondo termometro alla stessa ora. La funzione deve restituire una nuova lista contenente, per ogni copia di misurazioni, il valore massimo registrato in città. La funzione deve essere implementata utilizzando le sole primitive dell’ADT lista.

Ad esempio, se `l1 = {7, 3, 6, 4}`, ed `l2 = {5, 2, 5, 6}`, la funzione restituirà la lista `{7, 3, 6, 6}`, poiché $7 > 5$, $3 > 2$, etc.

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>

int max = 2;

char * mescola(int step, char * v, char threshold){
    while(v[max+step]!='\0')
        if (v[max+step] > threshold)
            max++;
        else
            v[max+step] = *(v+max+step)+1;
    return &(v[step])-step;
}

int main(){
    char word[] = "messaggio";
    char * temp;

    temp = mescola(-2, word, 'c');
    printf("%s %d\n", temp, max);

    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 calcolandone il risultato secondo la rappresentazione binaria in complemento a 2 (si trasli anche il risultato in decimale per verificare la correttezza dell'operazione):

$$89 + (-40)$$

ESERCIZIO 5 (3 punti)

Si illustri cosa fanno e quale è la differenza fra `fprintf` e `fwrite`.

Supponendo che sia:

```
int x = 215;
```

Che differenza c'è fra:

```
fprintf(file, "%d", x);
```

e

```
fwrite(&x, sizeof(int), 1, file);
```

Cosa verrà scritto nel file?

Nota: il codice ASCII di '1' è 49 (in base 10); inoltre il candidato faccia riferimento ad una architettura "little endian", dove cioè i byte meno significativi sono a destra. Si supponga una rappresentazione degli interi su 4 bytes.

Soluzioni

ESERCIZIO 1

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

#define MAX_MISURE 120

typedef struct {
    char nome[32];
    int codice;
    int elenco[MAX_MISURE];
    int dim;
} pista;

int * seleziona(pista p[], int dim, int *dimResult) {
    int i, j, k;
    int *result;
    int countDays;
    result = (int*) malloc(sizeof(int) * dim);
    k = 0;
    for (i=0; i<dim; i++) {
        countDays = 0;
        for (j=0; j<p[i].dim; j++)
            if (p[i].elenco[j] > 100)
                countDays++;
        if (countDays > (p[i].dim)/3) {
            result[k] = p[i].codice;
            k++;
        }
    }
    *dimResult = k;
    return result;
}

void creaFile() { // NON RICHIESTA NEL TESTO !!!
    FILE * fp;
    int i;
    int j;
    pista temp;
    fp = fopen("valori.dat", "wb");
    for (i=0; i<100; i++) {
        sprintf(temp.nome, "nome %d", i);
        temp.codice = i;
        for (j=0; j<i; j++)
            temp.elenco[j] = 90 + j;
        temp.dim = j;
        fwrite(&temp, sizeof(pista), 1, fp);
    }
    fclose(fp);
}

int main(void) {
    FILE *fp;
    int dim = 0;
    pista p[100];
    int * pisteBelle;
    int dimPisteBelle;
```

```

int i;

creaFile();

if ((fp=fopen("valori.dat", "rb")) == NULL)
    exit(1);
while (fread(&p[dim], sizeof(pista), 1, fp) == 1)
    dim++;
fclose(fp);
pisteBelle = seleziona(p, dim, &dimPisteBelle);
for (i=0; i<dimPisteBelle; i++)
    printf("%d\n", pisteBelle[i]);

system("PAUSE");
return (0);
}

```

ESERCIZIO 2

```

list selectMax(list l1, list l2) {
    if (empty(l1) || empty(l2))
        return emptylist();
    else {
        if (head(l1) > head(l2))
            return cons(head(l1), selectMax(tail(l1), tail(l2)));
        else
            return cons(head(l2), selectMax(tail(l1), tail(l2)));
    }
}

```

ESERCIZIO 3

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

```
messdggio 11
```

La funzione `main()` si limita ad invocare la funzione `mescola(...)`, e poi a stampare il valore della variabile globale `max` e della stringa `temp` (restituita dall'invocazione della funzione `mescola(...)`).

La funzione `mescola` scorre attraverso un ciclo tutti i caratteri dell'array passato come parametro: in particolare se un carattere è "maggiore" (cioè segue alfabeticamente) il carattere passato tramite il parametro `threshold`, allora l'indice di accesso all'array viene incrementato. In caso contrario, al carattere viene sostituito il carattere che lo segue nell'alfabeto (si noti che in tal caso l'indice di accesso all'array non viene incrementato, perciò all'iterazione successiva si è ancora sullo stesso carattere). Siccome la `threshold` è il carattere 'c', e poiché l'unico carattere minore o uguale a 'c' è la lettera 'a', questa viene sostituita dapprima con 'b', poi con 'c', ed infine con 'd'.

Il valore della variabile globale `max` è pari al valore iniziale (2), più la lunghezza della stringa (9): quindi al termine `max` vale 11 (che è il valore stampato).

ESERCIZIO 4

```

89    ->    01011001          01011001 +   (89)
+40   ->    00101000          11011000 =  (-40)
                11010111          -----
-40   ->    11011000          00110001   (+49)

```