

Fondamenti di Informatica T-1 (A.A. 2009/2010) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 15 Luglio 2010 – durata 1h
Totale 12 punti, sufficienza con 7

ESERCIZIO 1 (6 punti)

Siano dati un vettore di interi `position` di lunghezza `length` ed una lista di char `input` di lunghezza qualsiasi. Si realizzi una funzione (ricorsiva)

```
list reorder(int* position, int length, list input);
```

che restituisca una nuova lista con gli elementi di `input`, selezionandone gli elementi in base al valore delle celle del vettore `position`. In particolare, bisogna scansionare il vettore `position` (partendo dalla prima cella) ed inserire nella lista restituita l'elemento della lista `input` con posizione corrispondente al valore della cella attualmente considerata (1 corrisponde al primo carattere, 2 al secondo e così via); se la posizione indicata non è valida, la funzione deve restituire la lista realizzata fino a quel momento.

Ad esempio se `position = [3, 2, 4, 9, 2]` e `input = ['a', 'b', 'e', 'f', 'h']`, la funzione `reorder` restituisce la lista `['e', 'b', 'f']` poiché 'e' è in posizione 3, 'b' in posizione 2, 'f' in posizione 4 e 9 è superiore al numero di elementi della lista `input` (e quindi la funzione `reorder` restituisce la lista creata). Notare che la funzione `reorder` termina anche quando tutte le celle dell'array `position` sono state scansionate.

La funzione `reorder` dovrà essere implementata utilizzando le sole primitive dell'ADT lista; ogni altra funzione dovrà essere opportunamente specificata dal candidato. Si realizzi inoltre un semplice `main` di prova che invochi correttamente la funzione `reorder` creata.

Nota: l'ordine degli elementi della lista restituita dalla funzione `reorder` è ininfluente.

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione:

```
char sum_char(char input, int shift){
    input += shift;
    if(input >= 'g')
        return input-1;
    else
        return sum_char(input, shift+1.7);
}
```

Mostrare la sequenza dei record di attivazione ed il valore di ritorno nel caso in cui la funzione sia invocata con parametri attuali ('a',1).

ESERCIZIO 3 (3 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>

int i = 0;

char* merge(char* v1, int n1, char* v2, int n2){
    char *res, *temp;
    temp = res = (char*) malloc(sizeof(char)*(n1+n2));
    while( *v1!='\0' && *v2!='\0'){
        if(*v1 < *v2){
            *temp = *v1;
            v1++;
        }
        else {
            *temp = *v2;
            v2++;
        }
        temp++;
        i++;
    }
    return res;
}

int main(){
    char* s1 = "mare";
    char* s2 = "ape";
    char* fin;
    int k, l1 = 0, l2 = 0;
    while(*(s1+l1) != '\0') l1++;
    while(*(s2+l2) != '\0') l2++;
    fin = merge(s1, l1, s2, l2);
    printf("%d\n", i);
    for(k=0; k<i; k++){
        printf("%c", fin[k]);
    }
    printf("\n");
    return 0;
}
```

ESERCIZIO 4 (1 punto)

Il candidato illustri brevemente le differenze, anche in termini di occupazione di spazio disco, fra l'utilizzo di un file di testo o di un file binario per la memorizzazione di una sequenza di dati numerici di tipo int.

Soluzioni

ESERCIZIO 1

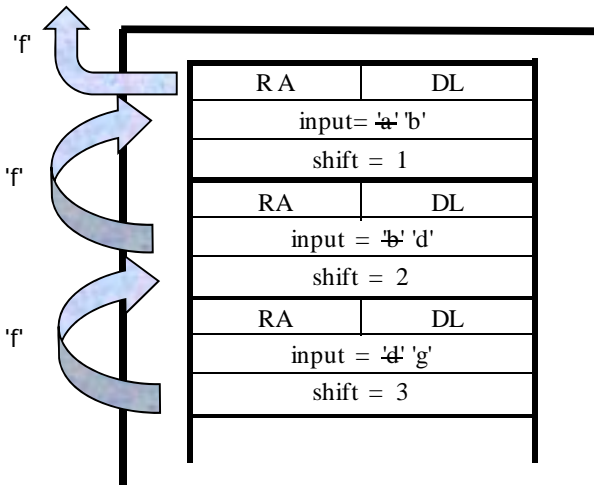
```
#include "list.h"
#include "element.h"

list reorder(int* position, int length, list input){
    int i;
    list temp;
    if(length <= 0 ) return emptylist();
    else{
        temp = input;
        for(i=1; i<*position && !empty(temp); i++){
            temp = tail(temp);
        }
        if(empty(temp)) return emptylist();
        else return cons(head(temp), reorder(position+1, length-1, input));
    }
}

int main(){
    list input, res;
    int length = 5;
    int position[] = {3, 2, 4, 9, 2};
    input = cons('a', cons('b', cons('e', cons('f', cons('h', emptylist()))));
    res = reorder(position, length, input);
    while(!empty(res)){
        printf("%c\n", head(res));
        res = tail(res);
    }
    return 0;
}
```

ESERCIZIO 2

La funzione restituisce il carattere 'f'.



ESERCIZIO 3

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

```
5  
amape
```

La funzione `main()` calcola la lunghezza delle stringhe `s1` ed `s2` ed invoca la funzione `merge()`.

La funzione `merge()` alloca spazio sufficiente per contenere tutti i caratteri delle stringhe `v1` e `v2`, ma senza considerare i terminatori. Il ciclo `while` scansiona i caratteri di tali vettori partendo dalla prima cella ed inserisce nello spazio di memoria allocato dinamicamente il carattere attualmente considerato tra `v1` e `v2` che sia minore da un punto di vista lessicografico. Il ciclo `while` termina non appena uno dei due vettori è stato scansionato fino alla fine. La funzione `merge()` restituisce al chiamante un riferimento alla prima cella dell'area di memoria allocata dinamicamente.

La funzione `main()` scrive sullo standard output il numero di caratteri scritti nell'area di memoria allocata dalla funzione `merge()` e l'elenco dei caratteri effettivamente scritti.