

Fondamenti di Informatica T-1 (A.A. 2014/2015) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 29 Gennaio 2015 – durata 1h
Totale 12 punti, sufficienza con 7

Compito A

ESERCIZIO 1 (6 punti)

Sono date due liste di interi `l1` e `l2`. La lista `l1` contiene interi positivi mentre `l2` è una lista ordinata di interi non negativi. Si realizzi la funzione ITERATIVA

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

che estragga gli elementi della lista `l1` nelle posizioni indicate dagli elementi nella lista `l2` (il valore 0 indica il primo elemento, il valore 1 indica il secondo elemento...). Qualora `l2` contenga indici per posizioni non esistenti di `l1`, dovrà essere inserito il valore -1. Per esempio, se `l1 = {4, 42, 11, 29, 69, 17}`, `l2 = {1, 4, 10, 12}`, la funzione `estrai()` deve restituire la lista `l3 = {42, 69, -1, -1}`.

La funzione `estrai()` dovrà essere implementata utilizzando le primitive dell'ADT lista. Si realizzi inoltre una semplice funzione `main()` di prova che invochi correttamente la funzione `estrai()` creata.

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione

```
double f(int x){
    if (x==0){
        return 0;
    }else
        if(x==1){
            return 1;
        }else{
            return (f(x-1.0) + f(x-2.0));
        }
}
```

Mostrare la sequenza dei record di attivazione ed il valore di ritorno nel caso in cui la funzione sia invocata con parametri attuali (3).

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).

Nota: si ricorda che l'operazione del modulo è il resto della divisione tra interi.

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

char* conv(int n,int *esito){

    int i;
    char *b;

    b = (char*)malloc(5*sizeof(char));
    *(b+4)='\0';

    for(i=3;n!=1;i--){
        if(i==0){
            *esito=1;
            return b;
        }else{
            *(b+i)=(n%2)+'0';
            n=n/2;
        }
    }
    b[i--]='1';
    for(;i>=0;i--){
        b[i]='0';
    }
    *esito=0;
    return b;
}

int main(){

    int a=5;
    int c;
    char *r;

    r= conv(a,&c);

    printf("%d\n",c);
    if(c==0){
        printf("%s\n",r);
    }

    return 0;
}
```

ESERCIZIO 4 (1 punto)

Sono stati visti due modi diversi per rappresentare i numeri interi con segno in forma binaria: rappresentazione in modulo e segno e rappresentazione in complemento a due. Se ne illustrino brevemente le differenze e i rispettivi vantaggi e svantaggi.

Soluzioni

ESERCIZIO 1

```
list estrai(list l1, list l2){

    int i=0;
    list temp = emptyList();
    list res = emptyList();

    while(!empty(l2)){
        if(empty(l1)){
            temp=cons(-1,temp);
            l2=tail(l2);
        }else{
            if(head(l2)==i){
                temp=cons(head(l1),temp);
                l2=tail(l2);
            }
            l1=tail(l1);
            i++;
        }
    }

    // la lista temp viene costruita "al contrario", ora la ribaltiamo per
    avere gli elementi della lista risultato nelle posizioni corrispondenti a quelle
    degli elementi di l2.
    while(!empty(temp)){
        res=cons(head(temp),res);
        temp=tail(temp);
    }
    return res;
}

int main(){

    list valori,indici,risultato;

    valori=cons(4,cons(42,cons(11,cons(29,cons(69,cons(17,emptyList())))));
    indici=cons(1,cons(4,cons(10,cons(12,emptyList()))));

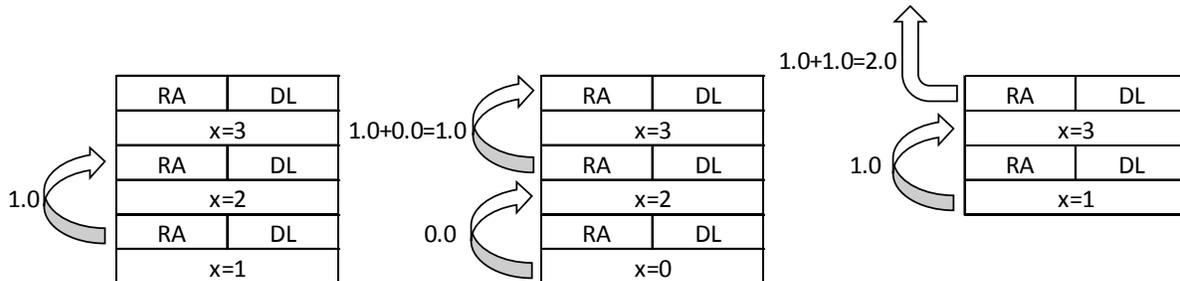
    risultato = estrai(valori,indici);

    while(!empty(risultato)){
        printf("%d\n",head(risultato));
        risultato=tail(risultato);
    }

    return 0;
}
```

ESERCIZIO 2

La funzione restituisce il valore 2.



ESERCIZIO 3

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

```
0
0101
```

La funzione `main()` inizializza una variabile intera `a` con valore 5 e dichiara due variabili: un intero `c` e un puntatore a carattere `r`. Invoca poi la funzione `conv()`, passando come argomento le variabili `a` e il riferimento a `c` e assegnando il risultato alla variabile `r`.

La funzione `conv()` dichiara una variabile intera `i` e un puntatore a carattere `b`. Questo viene fatto puntare ad un'area di memoria allocata dinamicamente dimensionata per contenere 5 valori `char` e assegna immediatamente il terminatore di stringa `'\0'` alla posizione `b[4]`, ovvero l'ultima. Esegue un ciclo `for` usando la variabile `i` come indice e iterando partendo dal valore 3 decrementandolo di uno fino a che l'argomento `n` della funzione è diverso da 1. Ad ogni passo verifica che la variabile `i` sia uguale a 0. In caso positivo, assegna alla variabile referenziata dal puntatore a `int esito`, argomento della funzione, il valore 1 e ritorna il puntatore a carattere `b`. Altrimenti, assegna alla posizione `i`-esima della stringa `b` il carattere risultante dalla somma tra il carattere individuato dal risultato dell'operazione modulo tra `n` e il valore 2 e il carattere `'0'`. Successivamente, assegna a `n` il risultato della divisione intera tra il suo attuale valore e il numero 2.

Terminato il ciclo, assegna alla posizione `i`-esima della stringa `b` il carattere `'1'`. L'accesso alla cella `i`-esima avviene con un'operazione di post-decremento sulla variabile `i`, aggiornandone il valore dopo che è stata usata per l'indirizzamento.

Si ha quindi un secondo ciclo `for`, privo di istruzione di inizializzazione, che continua a decrementare di 1 il valore di `i` ad ogni passo e si ripete finché la variabile `i` è pari o superiore a 0. In questo ciclo, ad ogni passo viene inserito nella posizione `i`-esima della stringa `b` il carattere `'0'`. Terminato, assegna alla variabile referenziata dal puntatore a `int esito`, argomento della funzione, il valore 0 e ritorna il puntatore a carattere `b`.

La funzione `main()` stampa sullo standard output il valore contenuto dalla variabile `c` (0) e, se `c` vale 0, la stringa indirizzata dalla variabile `r` (0101).