

Fondamenti di Informatica T-1 (A.A. 2018/2019) - Ingegneria Informatica
Prof.ssa Mello
Prova Scritta – 11 Luglio 2019 – durata 1h
Totale 12 punti, sufficienza con 7

Compito A

ESERCIZIO 1 (6 punti)

Si scriva una funzione **RICORSIVA**

```
int LimitList(list List1, int Min, int Max)
```

in cui `List1` è una lista di elementi interi, e che restituisca risultato vero se `List1` ha solo elementi compresi fra `Min` e `Max` (`Min` e `Max` inclusi, dove $Min \leq Max$). `LimitList` restituisce vero anche se la lista `List1` è vuota.

Si realizzi una funzione `main()` che utilizzi correttamente la funzione `LimitList`, leggendo da input i valori di `Min` e `Max`, chiamando la funzione `LimitList` con la lista `List1 = [9, 6, 3, 4]` e stampando a terminale la stringa vero o falso in base al suo risultato.

Si noti che se `Min` e `Max` fossero rispettivamente 2 e 9 la risposta sarebbe "vero", mentre se fossero 3 e 7 la risposta sarebbe "falso".

La funzione dovrà essere implementata utilizzando le primitive dell'ADT lista, includendo "list.h".

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione

```
double M(int a){
    if ( a >= 15 )
        return a;
    else {
        return M(a*2) + M(a*3);
    }
}
```

mostrare la sequenza dei record di attivazione ed il valore di ritorno nel caso in cui la funzione sia invocata con parametro attuale 5

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 ff(int x[], int n, char *y){
    int i = n-1;
    while (i>0) {
        if ((n % 2) != (x[--i]%2)) {
            n = x[i];
            (*y)++;
        }
        else {
            printf("%c\n", *y);
        }
        i--;
    }
}

int main(){
    int x[]={1,7,40,2,6,7,8,113,4,10,12}, n = 9;
    char y = 'a';
    ff(x, n, &y);
    printf("%c\n", y);
    return 0;
}
```

ESERCIZIO 4 (1 punto)

Si discutano le principali similitudini e differenze tra le strutture dati pila e coda.

Soluzioni

ESERCIZIO 1

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

int LimitList(list List1, int Min, int Max) {
    if (empty(List1))
        return 1;
    else {
        if ((head(List1) > Max) || (head(List1) < Min))
            return 0;
        else
            return LimitList(tail(List1), Min, Max);
    }
}

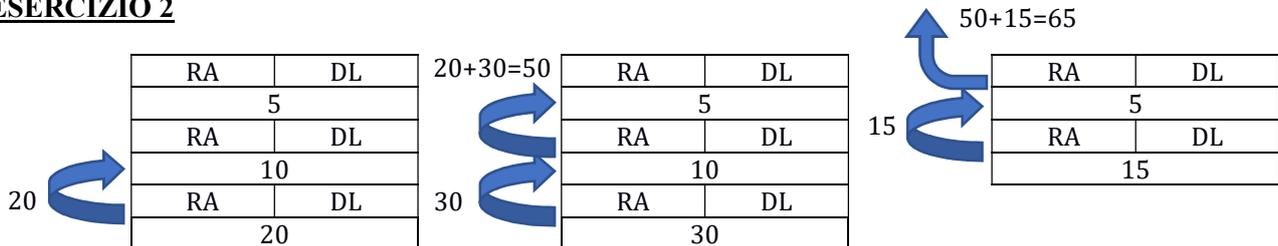
int main() {
    int n;
    int Min, Max;
    list list1=cons(9, cons(3, cons(6, cons(4, emptylist()))));

    printf("inserisci i valori di Min e Max con Min minore di Max");
    scanf("%d%d ", &Min, &Max);

    n= LimitList(list1,Min, Max);
    if (n == 1)
        printf("vero");
    else
        printf("falso");

    return 0;
}
```

ESERCIZIO 2



ESERCIZIO 3

Il programma è corretto e l'output prodotto è:

a
a
c

La funzione `main` inizializza un array `a` e invoca la funzione `ff` passando come parametri l'array `x`, l'intero `n` con valore 9 e il carattere 'a' per riferimento.

All'interno della funzione `ff` viene attribuito all'intero `i` il valore 8. Poiché $8 > 0$, si entra dentro il ciclo `while`. All'interno del ciclo, viene computato il modulo 2 di `n`. Essendo il valore 9, il risultato è 1.

La variabile `i` viene decrementata diventando 7. Viene selezionato l'intero in posizione 7 nel vettore `x`, ovvero 113, e ne viene calcolato il modulo 2, che da come risultato 1.

Poiché i due moduli sono uguali, allora si entra nel ramo `else`, dove viene stampato il carattere puntato da `y`, che è 'a'.

Infine la variabile `i` viene decrementata a 6.

Poiché $6 > 0$, avviene un altro ciclo.

All'interno del ciclo, viene di nuovo computato il modulo 2 di `n`. Essendo il valore 9, il risultato è 1.

La variabile `i` viene decrementata diventando 5. Viene selezionato l'intero in posizione 5 nel vettore `x`, ovvero 7, e ne viene calcolato il modulo 2, che da come risultato 1.

Poiché i due moduli sono uguali, allora si entra nel ramo `else`, dove viene stampato il carattere puntato da `y`, che è 'a'.

Infine la variabile `i` viene decrementata a 4.

Poiché $4 > 0$, avviene un altro ciclo.

All'interno del ciclo, viene di nuovo computato il modulo 2 di `n`. Essendo il valore 9, il risultato è 1.

La variabile `i` viene decrementata diventando 3. Viene selezionato l'intero in posizione 3 nel vettore `x`, ovvero 2, e ne viene calcolato il modulo 2, che da come risultato 0.

Poiché i due moduli sono diversi, allora `n` assume il valore 2 e la variabile puntata da `y` viene incrementata, diventato 'b'.

Infine la variabile `i` viene decrementata a 2.

Poiché $2 > 0$, avviene un altro ciclo.

All'interno del ciclo, viene di computato il modulo 2 di `n`. Essendo il valore 2, il risultato è 0.

La variabile `i` viene decrementata diventando 1. Viene selezionato l'intero in posizione 1 nel vettore `x`, ovvero 7, e ne viene calcolato il modulo 2, che da come risultato 1.

Poiché i due moduli sono diversi, allora `n` assume il valore 7 e la variabile puntata da `y` viene incrementata, diventato 'c'.

Infine la variabile `i` viene decrementata a 0.

Poiché non è vero che $0 > 0$, il ciclo termina e termina anche la funzione `ff`.

Nella funzione `main`, viene stampata la variabile `y`. Poiché era stata passata per riferimento, le modifiche avvenute nella funzione `ff` sono mantenute, e perciò stampato è 'c'.

NOTA: La funzione `ff(...)` non ha l'istruzione di `return`. Il programma viene compilato lo stesso, e il compilatore si limita solo a segnalare un warning a riguardo.