Fondamenti di Informatica T-1 (A.A. 2014/2015) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 9 Luglio 2015 – durata 1h
Totale 12 punti, sufficienza con 7

ESERCIZIO 1 (6 punti)

Data una lista di `float` di nome `l` e un `float` di nome `max`, si realizzi una funzione RICORSIVA

```
list extract(list l, float max);
```

che restituisca una nuova lista contenetene gli elementi della lista `l` che differiscono al più `max` dall'elemento successivo, considerando la differenza in valore assoluto. Inoltre, l'ultimo elemento della lista `l` deve sempre essere inserito nella lista restituita. Ad esempio, se `l = [3.2, 4.9, -1.1, -2.7, 5.3]` e `max = 3.1`, la funzione `extract()` restituisce `[3.2, -1.1, 5.3]` in quanto la differenza (3.2 - 4.9) in valore assoluto è inferiore a 3.1, la differenza (-1.1 - -2.7) in valore assoluto è inferiore a 3.1 e 5.3 è l'ultimo elemento.

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

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione:

```
int fun(char ch, int plus){
      int x;
      if ( ch > 'e' )
            return plus * 0.5;
      else{
            x = ch + plus;
            return fun(x, plus + 2.8);
      }
}
```

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

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

int* add(int*x, int b[], int c){
	int i;
	int *res = x;
	for(i=0; i < c; i++, x++){
		*x = *x + *(b+i);
	}
	return res;
}

int main(){
	int i;
	int x[] = {1,2,3,4};
	int y[] = {-1,2,-3,4};
	int *z;
	z = add (x, y, *(x+2) );
	for(i=0;i<4;i++){
		printf("%d %d\n", x[i], z[i]);
	}
	return 0;
}
```

ESERCIZIO 4 (1 punto)

Il candidato illustri brevemente cosa si intende per **variabile globale**, fornendone un esempio concreto.

Soluzioni

ESERCIZIO 1

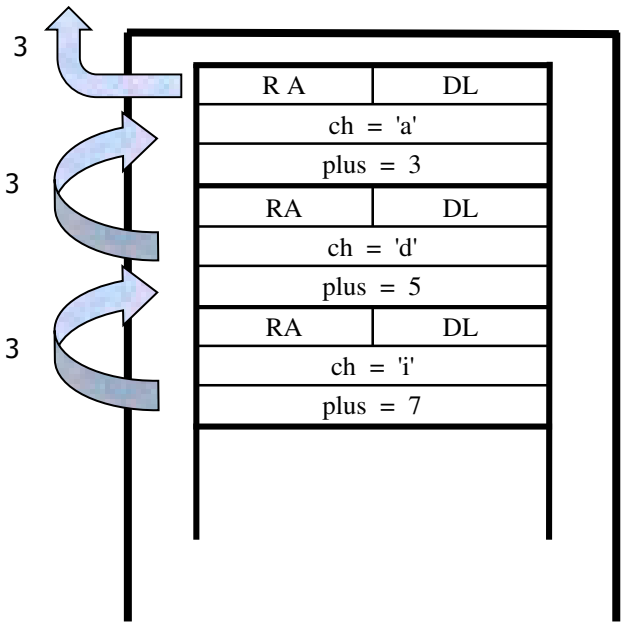
```
list extract(list l, float max){
      float diff;
      if(empty(l)) return l;
      else if( empty(tail(l)) ) return l;
      else {
            diff = head(l) - head(tail(l));
            if ( diff < 0 ) diff = -diff;
            if( diff <= max ) return cons( head(l), extract(tail(l), max));
            else return extract(tail(l), max);
      }

}

int main(){
      list l, res;
      l = cons(3.2, cons(4.9, cons(-1.1, cons(-2.7, cons(5.3, emptylist() )))));
      res = extract(l, 3.1);
      while( !empty(res) ){
            printf("%f\n", head(res));
            res = tail(res);
      }
      return 0;
}
```

ESERCIZIO 2

La funzione restituisce il valore intero 3.



ESERCIZIO 3

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

```
0 0
4 4
0 0
4 4
```

La funzione `main()` invoca la funzione `add()` con parametri di ingresso due array di interi e un intero con valore 3.

La funzione `add()` utilizza un ciclo `for` per iterare lungo i primi 3 elementi interi delle aree di memoria puntate dalle variabili `x` e `b`. Ogni interazione del ciclo somma l'i-esimo elemento presente nell'area di memoria referenziata da `b` con il corrispondente elemento presente nell'area di memoria referenziata in `x` (notare l'istruzione `x++` del `for` che permette di avanzare lungo gli elementi di `x`) e modifica il valore di quest'ultimo. Infine la funzione `add()` restituisce un riferimento al primo elemento dell'area di memoria precedentemente referenziata da `x`.

La funzione `main()` stampa sullo standard output i 4 valori di `x` e `z`, identici in quanto le due variabili puntano alla stessa area di memoria.