Fondamenti di Informatica T-1 (A.A. 2010/2011) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Mercoledì 16 Febbraio 2011 – durata 1h
Totale 12 punti, sufficienza con 7
Compito B

ESERCIZIO 1 (6 punti)

Date due liste di float `valori` e `soglie` (liste non vuote) ed un float `maxMedia`, si realizzi una funzione ricorsiva

```
list medie(list valori, list soglie, float maxMedia);
```

che restituisca una nuova lista contenente per ciascun elemento di `soglie` la media degli elementi contenuti nella lista `valori` scartando tutti quegli elementi con valore superiore all'elemento della lista `soglie` attualmente in esame; tale media deve essere inserita nella lista restituita solo se il suo valore è inferiore al valore `maxMedia`.

Ad esempio se `valori = [4.0, -2.0, 7.0, 1.0]`, `soglie = [0.0, 5.0]` e `maxMedia = 0.5`, la funzione `medie()` deve restituire la lista `[-2.0]` poiché per il primo elemento di `soglie` la media è -2.0 (ovvero -2/1) che è inferiore a 0.5, mentre per il secondo elemento di `soglie` la media è 1.0 (ovvero (4-2+1)/3) che è superiore a 0.5.

La funzione `medie()` dovrà essere implementata utilizzando le sole 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 `medie()` creata.

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

ESERCIZIO 2 (2 punti)

Si consideri la grammatica G con scopo S e simboli terminali {x, y, z, 7, 8, 9}

```
S ::= B Q | A R
A ::= R | R Q
B ::= Q B A | Q
Q ::= 7 | 8 | 9
R ::= x | y | z
```

La stringa "87z9" appartiene al linguaggio generato da tale grammatica? In caso affermativo, se ne mostri la derivazione left-most.

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>

char* atMostDistant(char* chars, int* delta){
	int dim=0, i=0;
	char *res;
	char *temp = chars;
	while( *temp != '\0' ){
		dim++;
		temp++;
	}
	dim++;

	res = (char*)malloc(sizeof(char)*dim);

	while( ! ( *chars=='\0' || *(chars+1)=='\0' ) ){
		if( *(chars+1) - *chars <= *delta ){
			res[i] = *chars;
			i++;
		}
		else{
			printf("scarto %c\n", *chars);
		}
		chars++;
		delta++;
	}

	res[i] = '\0';
	return res;
}

int main() {
	char v[] = "bchp";
	int t[] = {9, 3, 2, 3};
	char *res;

	res = atMostDistant(v, t);

	printf("%s", res);
	return 0;
}
```

ESERCIZIO 4 (1 punti)

Il candidato illustri brevemente le varie fasi che consentono di ottenere un programma eseguibile partendo da codice sorgente.

Soluzioni

ESERCIZIO 1

```
list medie(list valori, list soglie, float maxMedia){
      float count, i, mediaCorrente;
      list tempValori;
      if( empty(soglie) ){
            return emptylist();
      }
      else{
            count = 0;
            i = 0;
            tempValori = valori;
            while( ! empty(tempValori) ){
                  if( head(tempValori) <= head(soglie) ){
                        count += head(tempValori);
                        i++;
                  }
                  tempValori = tail(tempValori);
            }
            mediaCorrente = count / i;
            if( mediaCorrente < maxMedia ){
                return cons( mediaCorrente, medie(valori, tail(soglie), maxMedia));
            }
            else{
                  return medie(valori, tail(soglie), maxMedia);
            }
      }
}

int main(){
      list valori, soglie, res;
      float maxMedia = 0.5;
      valori = cons(4, cons(-2, cons(7, cons(1, emptylist()))));
      soglie = cons(0, cons(5, emptylist()));
      res = medie(valori, soglie, maxMedia);
      while( ! empty(res) ){
            printf("%f\n", head(res));
            res = tail(res);
      }
      return 0;
}
```

ESERCIZIO 2

La frase appartiene al linguaggio. In particolare, la si può ottenere tramite la seguente derivazione left-most:
S → BQ → QBAQ → 8BAQ → 8QAQ → 87AQ → 87RQ → 87zQ → 87z9

ESERCIZIO 3

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

```
scarto c
scarto h
b
```

La funzione `main()` inizializza le stringhe ben formate `v` e `t` e poi invoca la funzione `atMostDistant()`.

La funzione `atMostDistant()` alloca dinamicamente spazio di memoria sufficiente a contenere tutti i caratteri della stringa ben formata `temp` (precedentemente inizializzata a `chars`), terminatore compreso, la cui lunghezza è stata calcolata dal primo ciclo `while`. Il ciclo `while` itera finché la stringa ben formata `chars` ha almeno due caratteri (terminatore non compreso). Ad ogni iterazione il primo carattere della stringa `chars` viene aggiunto allo spazio di memoria allocato dinamicamente solo se la distanza tra il primo ed il secondo carattere della stringa `chars` è inferiore o uguale all'intero presente nella prima cella di memoria dell'array `delta`. Invece, se la distanza è superiore allora viene stampato un messaggio che indica che il carattere della stringa `chars` viene scartato. Alla fine di ogni ciclo `while` viene incrementato il valore sia di `chars` che di `delta`. Infine la funzione `atMostDistant()` inserisce il terminatore di stringa nella cella successiva all'ultimo carattere inserito nell'area di memoria allocata dinamicamente e restituisce un riferimento alla prima cella di tale area di memoria.

La funzione `main()` scrive sullo standard output la stringa ben formata restituita dalla funzione `atMostDistant()`.