Fondamenti di Informatica T-1 (A.A. 2011/2012) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 14 Giugno 2012 – durata 1h
Totale 12 punti, sufficienza con 7

ESERCIZIO 1 (6 punti)

Data una lista di float `l`, si realizzi una funzione RICORSIVA

```
list select(list l);
```

che restituisca una nuova lista contenente gli elementi della lista `l` i cui elementi hanno valore superiore alla somma di tutti gli elementi successivi al valore stesso.

Ad esempio se `l = [16, 2, 7, 3, 1, 1, 1]`, la funzione `select()` deve restituire la lista `[16, 7, 1]` in quanto 16 > 2 + 7 + 3 + 1 + 1 + 1, 7 > 3 + 1 + 1 + 1 e 1 > 0 (dove 1 è l'ultimo elemento della lista).

La funzione `select()` 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 `select()` creata.

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

ESERCIZIO 2 (2 punti)

Si consideri la grammatica G con scopo S, simboli non terminali {H, K, X, Y} e simboli terminali {a, b, c, d, e, f}:

```
S ::= HKX | KHY
H ::= XH | YK
K ::= KY | X
X ::= a | b | c
Y ::= d | e | f
```

La stringa "cdaf" 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>

char* purge(char* str, int* dim){

	char *res;
	int i;

	*dim = 0;
	while( *(str+*dim) != '\0' ){
		(*dim)++;
	}
	(*dim) += 1;

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

	for(i=*dim-2; i>=0; i--){
		if( str[i] > 'a'+3 ){
			res[i] = *(str+i);
		}
		else{
			res[i] = '-';
		}
	}
	res[*dim-1] = '\0';

	return res;
}

int main() {
	char* word = "adelaide";
	char *res;
	int n;
	res = purge(word, &n);
	printf("%d %s\n", n, res);
	return 0;
}
```

ESERCIZIO 4 (1 punti)

Il candidato illustri brevemente cosa è e a cosa serve la funzione `fopen()`, realizzando un breve esempio di codice che ne faccia uso correttamente.

Soluzioni

ESERCIZIO 1

```
list select(list l){
      list temp;
      float count = 0;
      if( empty(l) ){
            return emptylist();
      }
      else{
            temp = tail(l);
            while( ! empty(temp) ){
                  count += head(temp);
                  temp = tail(temp);
            }
            if( head(l) > count ){
                  return cons(head(l), select(tail(l)));
            }
            else{
                  return select(tail(l));
            }
      }
}

int main(){
      list l, res;
      l = cons(16, cons(2, cons(7, cons(3, cons(1, cons(1, cons(1,
                                                      emptylist()))))))));

      printf("select\n");
      res = select(l);
      while( ! empty(res) ){
            printf("%f ", head(res));
            res = tail(res);
      }
      printf("\n");

      return 0;
}
```

ESERCIZIO 2

La frase appartiene al linguaggio. In particolare, la si può ottenere tramite la seguente derivazione left-most:

S → KHY → XHY → cHY → cYKY → cdKY → cdXY → cdaY → cdaf

ESERCIZIO 3

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

```
9 --el-i-e
```

La funzione `main()` inizializza la stringa ben formata `word` e poi invoca la funzione `purge()`.

La funzione `purge()` conta il numero di caratteri della stringa ben formata `str` e poi alloca dinamicamente spazio di memoria sufficiente a contenere tutti i caratteri contati più il carattere di terminazione.

Il ciclo `for` itera lungo la stringa ben formata `str` partendo dall'ultimo carattere (escluso il terminatore di stringa) e copia i valori nello spazio di memoria allocato dinamicamente, con l'eccezione dei caratteri 'a', 'b', 'c', 'd' che vengono sostituiti col carattere '`-`'. Infine la funzione `purge()` inserisce il terminatore di stringa alla fine dell'area di memoria allocata dinamicamente e restituisce per valore un puntatore alla prima cella dell'area di memoria allocata dinamicamente e per riferimento la dimensione dell'area di memoria allocata dinamicamente.

La funzione `main()` stampa sullo standard output la dimensione dell'area di memoria allocata dinamicamente e i caratteri presenti nell'area di memoria allocata.