Fondamenti di Informatica T-1 (A.A. 2009/2010) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Lunedì 22 Dicembre 2009 – durata 1h
Totale 12 punti, sufficienza con 7
Compito B

ESERCIZIO 1 (6 punti)

Si supponga di avere a disposizione, già definite:

- l'ADT lista per interi (denominato **`list_int`**, con relative primitive **`emptylist_int(...)`**, **`cons_int(...)`**, **`head_int(...)`**, etc.).
- l'ADT lista per float (denominato **`list_float`**, con relative primitive **`emptylist_float(...)`**, **`cons_float(...)`**, **`head_float(...)`**, etc.).

Il candidato definisca una funzione *ricorsiva*:

```
list_float percentili(list_int l1, list_int l2);
```

La funzione deve restituire una nuova lista contenente i precentili dei valori contenuti in **`l2`** rispetto ad ognuno dei valori contenuti in **`l1`**, ognugno di questi ultimi intesi come soglie. Per "percentile" si intende il numero di elementi di **`l2`** pari o superiori ad una certa soglia **`el`**, diviso il numero totale di elementi memorizzati in **`l2`**. Si presti attenzione al fatto che se un elemento **`el`** di **`l1`**, inteso come soglia, risulta essere maggiore di tutti gli elementi di l2, allora il suo percentile è **`0`**. Si tenga conto anche del possibile caso in cui **`l2`** sia una lista vuota: in tal caso il percentile di qualunque elemento è sempre **`0`**. La funzione deve essere implementata utilizzando le sole primitive dell'ADT lista.

Al fine di semplificare la funzione di cui sopra, il candidato realizzi una funzione *iterativa*:

```
float calcolaPercentile(int el, list_int l);
```

che calcola il numero di elementi in **`l`** che sono maggiori o uguali al valore **`el`**di soglia, diviso il numero totale di elementi in **`l`**. Tale funzione deve essere implementata accedendo alle liste tramite la notazione a puntatore, e senza fare ricorso alle primitive dell'ADT.

Ad esempio, se **`l1 = {1, 4, 5}`**, ed **`l2 = {1, 3, 4, 1, 3, 1}`**, la funzione **`percentili(...)`** restituirà la lista **`{1, 0.166666, 0}`**, poiché in l2 ci sono 6 valori >= 1, su un totale di 6 elementi; un solo elemento è maggiore o uguale al valore 4, su 6 elementi (1/6 = 0.166666); poi non c'è nessun valore superiore alla soglia 5, e quindi il suo percentile vale 0.

ESERCIZIO 2 (2 punti)

Un elaboratore rappresenta i numeri interi su 8 bit tramite la notazione in complemento a 2. Indicare come viene svolta la seguente operazione aritmetica calcolandone il risultato secondo la rappresentazione binaria in complemento a 2 (si trasli anche il risultato in decimale per verificare la correttezza dell'operazione):

```
47 + (-57)
```

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 temp = 1;

int blip(char * v, char * p, int step){
      int i = step/2-2;
      int j = step/2;
      *(p+j-1) = v[j-1];
      for (;i>=0 && j<step-1;) {
            p[i] = v[j];
            p[j] = v[i];
            i--, j++;
      }
      *(p+step-1) = '\0';
    return i;
}

int main(){
      char * p;
      char str[] = "gateman";
      int dim=0;
      int temp;

      while(str[dim])
            dim++;
      dim++;
      p = (char *) malloc(sizeof(char)*dim);
      temp = blip(str, p, dim);
      printf("%s %d\n", p, temp);
      free(p);
      return (0);
}
```

ESERCIZIO 4 (1 punto)

Illustrare il concetto di funzione ricorsiva, presentando anche esempi di codice, e descrivere brevemente i vantaggi e gli svantaggi rispetto ad una corrispondente funzione iterativa.

Soluzioni

ESERCIZIO 1

```
float calcolaPercentile(int el, list_int l) {
      int dim = 0;
      int cont = 0;

      while (l != NULL) {
            if (l->value >= el)
                  cont++;
            dim++;
            l = l->next;
      }
      if (dim == 0)
            return 0;
      else
            return ((float) cont)/dim;
}

list_float percentili(list_int l1, list_int l2) {
      if (empty_int(l1))
            return emptylist_float();
      else
            return cons_float(
                        calcolaPercentile(head_int(l1), l2),
                        percentili(tail_int(l1), l2)
                     );
}
```

ESERCIZIO 2

```
47    ->    00101111                    00101111 +  (47)
+57   ->    00111001                    11000111 =  (-57)
            11000110                    ----------------
-57   ->    11000111                    11110110    (-10)

                                        11110110
                                        00001001
                                        ----------------
                                        00001010 -> (+10)
```

ESERCIZIO 3

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

`nametag -1`

La funzione `main()` dapprima calcola la lunghezza della stringa `str`, considerando nel conto anche il terminatore, e poi alloca memoria sufficente per contenere una stringa di lunghezza uguale. Quindi viene invacata la funzione `blip(…)`.

Tale funzione inizializza gli indici i e j ai due indici centrali dell'array v, escludendo l'elemento centrale (la stringa passata ha un numero dispari di carrateri). Nel caso particolare di una stringa di 7 caratteri, indici validi da 0 a 6, gli indici dei due caratteri attorno al carattere centrale sono 2 e 4. La funzione procede poi ad assegnare al vettore p i caratteri di v, scambiandoli pero' di posizione tramite i e j: il carattere all'indice 4 di v finisce all'indice 2 di p, il carattere all'indice 2 di v finisce all'indice 4 di p, e così via incrementando/decrementando rispettivametne le variabili i e j. Alla fine in p viene memorizzata la stringa str ma in ordine invertito. Un opportuno terminatore di stringa viene poi messo nell'array p nell'ultima posizione disponibile.

La funzione `main()` infine stampa a video la stringa p e l'intero ottenuto dalla funzione `blip(…)`.