Fondamenti di Informatica T-1 (A.A. 2013/2014) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 12 Giugno 2014 – durata 1h
Totale 12 punti, sufficienza con 7

ESERCIZIO 1 (6 punti)

Data una lista di stringhe ben formate `l1`, si realizzi una funzione RICORSIVA

```
list run(list l1);
```

che inserisca nella lista restituita tutti gli elementi che iniziano per `"NON"`.
Per esempio, se `l1 = ["N", "NONNA", "NONpappa", "amici"]`, la funzione `run()` deve restituire la lista `[NONNA, NONpappa]`, in quanto solo le stringhe `NONNA` e `NONpappa` iniziano per `NON`.

Si definisca inoltre la funzione

```
int iniziaCon(char *str)
```

che data in ingresso una stringa, verifichi se essa inizia o meno per `NON`.

La funzione `run()` dovrà essere implementata utilizzando le primitive dell'ADT lista e la funzione di libreria `strlen(char* str)` che restituisce la lunghezza di una stringa ben formata; ogni altra funzione dovrà essere opportunamente specificata dal candidato. Si realizzi inoltre una semplice funzione `main()` di prova che invochi correttamente la funzione `run()` creata.

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

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione:

```
int fun(char ch, int a){
      int m = ch - 'a'+ a;
      if (ch=='a' || m==9)
            return m*1.5;
      else
            return fun(--ch,a+2)*0.5;
}
```

Mostrare la sequenza dei record di attivazione ed il valore di ritorno nel caso in cui la funzione sia invocata con parametri attuali `('d', 2)`.

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 *fun(char *s1, char *s2){
      char *str, *a, *b;
      str = (char*)malloc(strlen(s1-1)*sizeof(char));
      a = b = str;

      while(*s1 != 'z'){
            if(*s2 == 'z'){
                  *a = '\0';
                  str = s2;
            }
            else{
                  *a = *s1;
                  str = a;
            }
            s1++;
            s2++;
            a++;
      }
      printf("%s\n", b);
      return str;
}

int main(void){
      char *s1 = "abcdz";
      char *s2 = "azvztvt";

      char *res = fun(s1, s2);

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

ESERCIZIO 4 (1 punto)

Il candidato illustri brevemente cosa sono e a cosa servono le funzioni C `fscanf()` e `fread()`, evidenziandone le principali differenze.

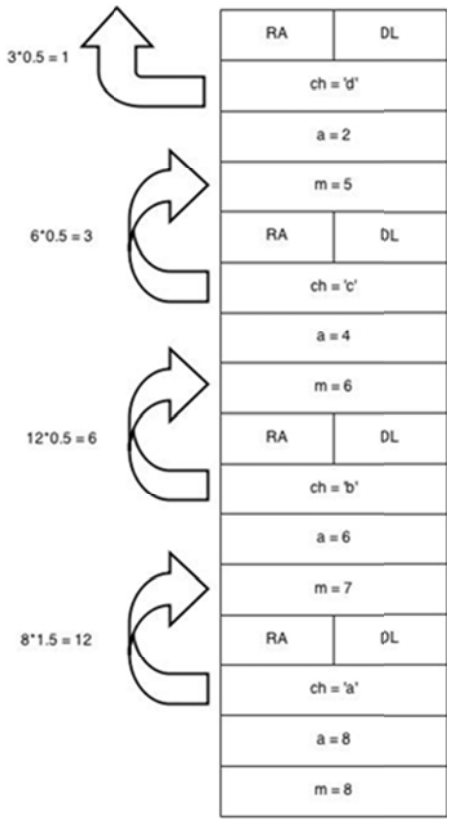
Soluzioni

ESERCIZIO 1

```
int iniziaCon(char *str)
{
      if( strlen(str) < 3 )
            return 0;
      else if(str[0] == 'N' && str[1] == 'O' && str[2]=='N')
            return 1;
      else
            return 0;
}

list run(list l1){
      if(empty(l1))
            return emptylist();
      else{
            if(iniziaCon(head(l1)))
                  return cons(head(l1), run(tail(l1)));
            else
                  return run(tail(l1));
      }
}

int main(){
      list res;
      list l1 = cons("N",cons("NONNA",cons("NONpappa",cons("amici",
emptylist())))));

      res = run(l1);

      while(!empty(res)){
            printf("%s \n",head(res));
            res = tail(res);
      }
      return 0;
}
```

ESERCIZIO 2

La funzione restituisce il valore 1.



ESERCIZIO 3

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

```
a
ztvt
```

La funzione `main()` inizializza due puntatori a carattere `s1` e `s2` e poi invoca la funzione `fun()`.

La funzione `fun()`alloca dinamicamente spazio di memoria sufficiente a contenere tanti char quanti sono gli elementi dell'array `s1` diminuiti di 1. Due ulteriori puntatori a carattere `a` e `b` sono inizializzati facendoli puntare alla spazio di memoria allocato dinamicamente per la variabile `str`. Successivamente il ciclo while itera sugli elementi dell'array `s1` finché ciascun valore è diverso dal carattere `'z'`.
Se il carattere attualmente puntato da `s2` è uguale a `'z'`, all'area di memoria allocata dinamicamente viene assegnato il carattere di fine stringa `'\0'` e si fa puntare `str` alla cella di memoria puntata da `s2`.
Se invece il carattere attualmente puntato da `s2` è diverso da `'z'`, all'area di memoria allocata dinamicamente viene assegnato il valore del carattere puntato da `s1` e si fa puntare `str` alla cella di memoria puntata da `a`. All'uscita del ciclo `while`, si stampa la stringa `b` che punta alla prima cella dell'area di memoria allocata dinamicamente. La funzione `printf` tronca la stampa al primo carattere di fine stringa trovato.
Infine la funzione `fun()`restituisce al chiamante un riferimento a `str` che all'uscita del `while` punta alla cella di memoria puntata da `s2`.

La funzione `main()` stampa sullo standard output il valore della variabile `res`.