Fondamenti di Informatica - A.A. 2012-2013

Ingegneria Informatica
Facoltà di Ingegneria dell'Informazione
Prof.ssa Cristiana Bolchini
Appello del **05/02/2012**

RESTITUIRE COMPILATO

Cognome	**Nome**	**Matricola**	**Voto:** .../**30**

Quesito:	1	2	3	4	5	Tot.
Max:	6	6	6	6	6	30
Punti:						

Istruzioni:
- non è possibile consultare libri, appunti, la calcolatrice o qualsiasi dispositivo elettronico, né comunicare;
- si può scrivere con qualsiasi colore, anche a matita, ad eccezione del **rosso**.
- tempo a disposizione: 2h 00m

Stile del codice C:
- non è necessario inserire direttive `#include`;
- i commenti non sono necessari, ma potrebbero essere utili nel caso di errore;

Quesito 1 (6 punti) Punteggio ottenuto: .../6

Scrivere un sottoprogramma che ricevuto in ingresso il nome di un file di testo, trasmette al chiamate il numero di parole (sequenza di caratteri separato da spazi), il numero di frasi (una frase termina quando c'è un punto) e il numero di caratteri in esso presenti. Si assuma che ogni paragrafo (una o più frasi terminate da un a-capo) non abbia più di 300 caratteri. Due parole possono essere separate da più di uno spazio, non ci possono essere due punti adiacenti.

Esempio

```
Questa e' una frase di esempio in cui si nota che, per caso, ci possono essere piu' spazi.  Mentre, due
punti vicini no.
Il carattere precedente era un a-capo, ma nel paragrafo precedente c'erano due frasi:  va benissimo.
```

Risultato

```
parole:  38
frasi:  3
caratteri:  223
```

Quesito 2 (6 punti) Punteggio ottenuto: .../6

Scrivere un sottoprogramma `array2list` che ricevuto in ingresso un array di caratteri restituisce una lista di un tipo opportuno (dichiarare il tipo per la rappresentazione degli elementi della lista `elem_t`) in cui ogni elemento rappresenta il carattere ed il numero di volte in cui compare nell'array in ingresso. Realizzare il sottoprogramma `array2list`. I sottoprogrammi qua riportati si riferiscono - per semplicità - al caso di lista per la gestione di dati interi: si immagini di disporre del sottoprogramma equivalente per la gestione di tipi di dati anche diversi, in base alle esigenze.

```
/* restituisce il numero di elementi presenti nella lista h */
int * lunghezza(t_elem * h);
/* restituisce il numero di elementi presenti nella lista h con campo informazione pari a ch */
int * conta(t_elem * h, char ch);
/* crea un nuovo elemento con campo informazione ch e occorrenze pari a 1 e lo inserisce in testa alla lista h, */
 * restituendo la testa  */
t_elem * instesta(t_elem * h, char ch);
/* crea un nuovo elemento di campo informazione ch e occorrenze pari a 1 e lo inserisce in coda alla lista h, */
 * restituendo la testa  */
t_elem * inscoda(t_elem * h, char ch);
/* crea un nuovo elemento di campo informazione ch e occorrenze pari a 1 e lo inserisce ordinatamente nella lista h, */
 * restituendo la testa  */
t_elem * insordinato(t_elem * h, char ch);
/* cerca nella lista h un termine con campo informazione ch e se esiste restituisce il puntatore a tale termine
 * altrimenti restituisce NULL */
t_term * cerca(t_elem * h, char ch);
/* cerca nella lista h un termine con campo informazione ch e se esiste restituisce 1, 0 altrimenti */
int esiste(t_elem * h, char ch);
/* elimina dalla lista h un termine con campo informazione ch e restituisce la testa della lista */
t_term * del(t_elem * h, char ch);
```

```
/* svuota la lista h */
void svuotalista(t_elem * h);
```

Quesito 3 (6 punti) Punteggio ottenuto: . . . /6

Scrivere un sottoprogramma che ricevuta in ingresso una stringa crea e restituisce al chiamante una stringa contenente tutte e soli i caratteri maiuscoli presenti nella stringa iniziale. La dimensione della stringa risultante deve essere tale da contenere tutti e soli i caratteri maiuscoli.

Quesito 4 (6 punti) Punteggio ottenuto: . . . /6

Si vuole realizzare un programma in grado di rappresentare informazioni relative a figure geometriche semplici (segmento 'S', triangolo 'T', quadrato 'Q' e rettangolo 'R') e complesse (costituite da un insieme di più figure semplici, identificare da 'X'). Ciascuna figura *semplice* è caratterizzata dall'identificatore del tipo, e da un insieme di punti nello spazio piano che la definiscono. Ciascuna figura complessa è costituita da al più 10 figure semplici.

1. Si definiscano i tipi di dati necessari per memorizzare una figura semplice (`simplef_t`) e una figura complessa (`complexf_t`).
2. Si scriva un sottoprogramma che, ricevendo come parametri una figura complessa, il numero di figure semplici che la costituiscono e il nome di un file, memorizzi i dati della figura complessa sul file.
3. Si discuta sull'opportunità di utilizzare un file in modalità binaria piuttosto che testuale (e si adotti la forma più opportuna).

Quesito 5 (6 punti) Punteggio ottenuto: . . . /6

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

int main(int argc, char * argv[]) {
   if (fork() == 0) {
      if (fork() == 0) {
         printf("3");
      } else if ((wait(NULL)) > 0) {
            printf("2");
      }
   } else {
      if (fork() == 0) {
            printf("1");
            exit(0);
      }
      if (fork() == 0) {
            printf("4");
         sleep(5);
      }
   }
   printf("0");
   return 0;
}
```

Dato il codice riportato accanto, quali delle seguenti sequenze possono essere generate dall'esecuzione del programma
(cancellare quelle non plausibili)?

a. 32010400

b. 31042000

c. 10403002

d. 43210000

e. 40302001

f. 14030200

g. 01302040

h. 30021004

Complessivamente, quanti processi vengono generati dal programma? _____

Quanti processi vengono generati dal processo iniziale? _____