



Dipartimento di Elettronica e Informazione

Politecnico di Milano

Prof. Luca Mottola

20133 Milano (Italia)

Piazza Leonardo da Vinci, 32

Tel. (39) 02-2399.3400

Fax (39) 02-2399.3411

Fondamenti di Informatica (ELT)

2 Febbraio 2015

Cognome

Nome

Matricola

Note

1. La mancanza dei dati di cui sopra comporta l'annullamento della prova.
2. Restituire **solo** queste pagine. Verranno ignorati fogli aggiuntivi. **Non** si può scrivere con la matita.
3. È proibito l'uso di dispositivi elettronici (computers, calcolatrici, cellulari, macchine fotografiche,...).
4. Sebbene non obbligatorio, si consiglia di inserire commenti nel codice per facilitarne la correzione.
5. Non si può tenere una copia del testo dell'esame quando si lascia l'aula.
6. Un punteggio inferiore a 8 punti preclude la registrazione del voto complessivo anche se sufficiente.
7. Coloro che hanno superato la prima prova con debito, devono colmare il debito completamente.
8. Il tempo a disposizione è di 2 ore.

Domanda 1: Puntatori e memoria dinamica (2 punti)

Si consideri la seguente funzione `boo()`, che opera sulla struttura dati `my_record` facendo uso di memoria dinamica. L'implementazione di `boo()` contiene un bug, dovuto ad un uso improprio della memoria dinamica attraverso i puntatori. Individuare tale bug e spiegarne, in un massimo di 80 parole, la natura e le problematiche che può comportare.

```
typedef struct {  
    int a;  
    int b;  
} my_record;  
  
int boo() {  
  
    my_record *p;  
    my_record *q;  
  
    p = malloc (sizeof(my_record));  
    q = p;  
    free(p);  
    q->a++;  
  
    return q->a;  
}
```


Domanda 2: Processi (3 punti)

Dato il seguente frammento di codice C, si mostri cosa è visualizzato su schermo in tutte le sue possibili esecuzioni. Non è possibile fare alcuna assunzione sulla velocità di esecuzione dei singoli processi o sulla presenza di alcuna sincronizzazione tra essi.

```
//...

int a = 0;
printf ("Hello World!\n");

if (fork() == 0) {
    a++;
    printf ("La variabile a vale %d\n", a);
} else {
    a--;
    printf ("La variabile a vale %d\n", a);
}
printf ("Fine!\n");

//...
```


Domanda 3: Ricorsione (4 punti)

Si vuole verificare se tutti gli elementi di una sequenza di interi positivi sono maggiori di un intero k .

- 1) Scrivere una funzione *ricorsiva* in linguaggio C che, ricevendo come parametri di input un array di n interi positivi e il valore di k , restituisca 0 se tutti gli elementi sono maggiori di k , oppure 1 altrimenti. Nel codice, si indichi chiaramente attraverso opportuni commenti quali sono i passi ricorsivi e quali sono i casi base della ricorsione. Non è possibile utilizzare variabili globali, ma la funzione può accettare un numero arbitrario di parametri.
- 2) Si simuli il funzionamento della funzione di cui sopra sulla sequenza $[7, 8, 9, 4]$ quando $k = 5$. In particolare, occorre illustrare la sequenza delle chiamate ricorsive con il valore attuale dei parametri ad ogni chiamata, e indicare il valore finale ritornato al codice chiamato.

Domanda 4: Strutture dati (6 punti)

Definire una struttura dati per rappresentare punti su un piano cartesiano; si supponga che le loro coordinate siano numeri interi.

Si supponga che il programma mantenga in memoria una lista dinamica, a singolo puntatore, di tali punti, ordinata come segue:

- dati due punti a e b , il punto a precede b nell'ordinamento se la sua distanza rispetto all'origine degli assi è minore rispetto a quella di b ;
- se a e b hanno la stessa distanza dall'origine, a precede b se l'ascissa di a precede l'ascissa di b ;
- se a e b hanno anche la stessa ascissa, a precede b se l'ordinata di a precede l'ordinata di b .

Vi è richiesto di:

- 1) scrivere una funzione che calcoli la distanza tra due punti, come valore intero;
- 2) scrivere una funzione che permetta di inserire un punto nella lista, mantenendo l'ordinamento;
- 3) scrivere una funzione che rimuova un punto date le sue coordinate;
- 4) scrivere una funzione che rimuova tutti i punti che si trovano alla stessa distanza da un punto p di riferimento.

In tutte queste funzioni, si supponga che la testa della lista NON sia una variabile globale. Si scrivano eventuali funzioni di supporto se opportuno.

