



Fondamenti di Informatica L-A Prof. Paolo Torroni

Seconda prova parziale 16 Dicembre 2005

Note allo svolgimento. Ti sono stati consegnati:

- a — il *testo del compito* (fronte: domande a risposta multipla, 15 punti; retro: esercizio di progetto, 18 punti);
- b — una *scheda anagrafica* con i tuoi dati: sul retro della scheda vi sono due adesivi che riportano un codice a barre;
- c — una *scheda delle risposte*;
- d — un *foglio di istruzioni* per la compilazione della scheda delle risposte.

Per un corretto svolgimento della prova, è necessario seguire in sequenza le seguenti istruzioni:

1. applicare *uno* dei due adesivi contenenti il codice a barre nell'apposito riquadro sulla scheda anagrafica;
2. scrivere il tipo di compito assegnato (**A**) sulla scheda anagrafica, di fianco al proprio nome, e scrivere proprio nome, cognome e matricola sul foglio protocollo che si utilizzerà per la soluzione dell'esercizio di progetto;
3. per la soluzione degli esercizi 1, 2, e 3 (domande a risposta multipla), bisogna compilare la scheda delle risposte, indicando sulla stessa con una croce × le risposte ritenute corrette, (vedi foglio di istruzioni). Per l'esercizio di progetto, bisogna utilizzare i fogli protocollo assegnati. Non verranno accettati compiti scritti a matita.
4. al termine della prova, applicare il secondo codice a barre adesivo nell'apposito riquadro sulla scheda delle risposte. Consegnare i due fogli su cui è stato applicato il codice a barre (scheda anagrafica e scheda delle risposte), e il foglio (o i fogli) protocollo con la soluzione dell'esercizio di progetto.

La prova va svolta individualmente e senza consultare libri, appunti, manuali, o strumenti elettronici (computer, cellulari, palmari ecc.). Nel complesso la prova dura 2 ore. La soglia per il superamento di questa prova è di 15 punti.

<p>1. [4]</p>	<p>Si considerino le funzioni nel linguaggio C:</p> <p>A Un file sorgente C, per essere compilato correttamente, deve contenere tutte le definizioni di tutte le funzioni che utilizza.</p> <p>B Una funzione non può restituire un tipo struct.</p> <p>C I parametri formali di una istanza di una funzione vengono inizializzati al valore dei parametri effettivi (attuali) all'atto della chiamata.</p> <p>D Si consideri una variabile <i>p</i>, di tipo puntatore a int, e una funzione <code>int f(int *Q)</code>, al cui interno <i>Q</i> viene modificato. La chiamata <code>f(p)</code>; modifica il valore di <i>p</i>.</p> <p>E Le istanze di funzioni utilizzano dei record di attivazione che vengono deallocati ogni volta che vengono generate ricorsivamente altre istanze della stessa funzione.</p>	
<p>2. [4]</p>	<p>Si considerino le caratteristiche del linguaggio C:</p> <p>A È possibile che in un programma vengano definite due variabili diverse aventi lo stesso identificatore, ed entrambe allocate nel data segment.</p> <p>B Lo stack può contenere variabili allocate dinamicamente (tramite la <code>malloc</code>).</p> <p>C Gli identificatori <code>stdin</code>, <code>stdout</code> e <code>stderr</code> denotano puntatori a FILE.</p> <p>D All'interno di una funzione è possibile definire blocchi di istruzioni contenenti nuove variabili, il cui campo d'azione diventa l'intero programma.</p> <p>E L'operatore di assegnamento tra variabili di tipo vettore è equivalente all'operatore di assegnamento tra variabili di tipo puntatore.</p>	<p>3. [7]</p> <p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> #define N 9 int f(int *a, int b) { if(*a < *(a+b)) /* punto 1 */ return b; else return *a=f(a, b+1)+b; } main() { int i, V[N]={0,1,2,3,4,5,6,7,8}; for (i=N-1; i>=0; i-=2) V[i]=(i+5)/2; /*punto 2 */ printf("%d\n\n", f(V,0)); /* punto 3 */ for (i=0; i<N; i++) /* punto 4*/ printf("%d\n", V[i]); } A L'istruzione prima del punto 2 (V[i]=(i+5)/2) viene eseguita 4 volte. B La funzione f è tail-ricorsiva. C L'istruzione prima del punto 1 (if(*a<*(a+b))) viene eseguita 3 volte. D L'istruzione al punto 3 stampa il valore 3. E Al punto 1 la variabile V non è visibile.</pre>