



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 (**D**) 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>Si considerino le funzioni nel linguaggio C:</p> <p>A I parametri effettivi (attuali) di una chiamata a funzione possono essere variabili allocate nello stack.</p> <p>B Due istanze diverse di una stessa funzione possono avere, all'interno del record di attivazione, lo stesso link dinamico (Dynamic Link).</p> <p>C I parametri formali di una funzione possono essere modificati all'interno della funzione stessa.</p> <p>D Il corpo di una funzione ricorsiva <i>f</i> può contenere anche due o più chiamate a <i>f</i>.</p> <p>E Lo stack contiene le variabili allocate dinamicamente tramite la <code>malloc</code>.</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> #define K 9 int f(int *x, int y) { if(*x < *(x+y)) /* punto 1 */ return y; else return *x=f(x,y+1)+y; } main() { int i, A[K]={0,1,2,3,4,5,6,7,8}; for(i=K-1; i>=0; i--=2) A[i]=(i+5)/2; /*punto 2 */ printf("%d\n\n", f(A,0)); /* punto 3 */ for (i=0; i<K; i++) /* punto 4*/ printf("%d\n", A[i]); } 3. [7]</pre> <p>A L'istruzione al punto 3 (<code>printf</code>) stampa il valore 6.</p> <p>B Al punto 1 la variabile <i>A</i> è visibile.</p> <p>C L'istruzione prima del punto 2 (<code>A[i]=(i+5)/2</code>) viene eseguita 5 volte.</p> <p>D Il ciclo al punto 4 stampa i seguenti valori (in linee separate): 3,1,3,3,4,5,5,7,6.</p> <p>E La funzione <i>f</i> è ricorsiva "tail".</p>
<p>Si considerino le caratteristiche del linguaggio C:</p> <p>A Un campo di una variabile di tipo <code>struct</code> può essere un vettore di puntatori.</p> <p>B È possibile tramite la <code>malloc</code> definire vettori di lunghezza non nota al momento della compilazione.</p> <p>C Tutte le variabili di un programma C sono contenute nello stack.</p> <p>D Tramite il qualificatore <code>static</code> è possibile definire variabili visibili anche al di fuori delle funzioni in cui sono definite.</p> <p>E Ogni variabile allocata dinamicamente (tramite la <code>malloc</code>) è posta nell'area heap.</p>	