



<p>1. [4]</p>	<p>Si considerino le funzioni nel linguaggio C:</p> <p>A [F] Lo stack contiene variabili di dimensione minore rispetto alle variabili contenute nello heap.</p> <p>B [V] I parametri formali vengono allocati nello stack.</p> <p>C [F] Un file sorgente C, per essere compilato correttamente, deve contenere tutte le definizioni di tutte le funzioni che utilizza.</p> <p>D [V] Dall'interfaccia (o prototipo) di una funzione è possibile ricavare il numero e tipo dei parametri da usare per invocarla correttamente.</p> <p>E [V] I parametri formali di una istanza di una funzione vengono inizializzati al valore dei parametri effettivi (attuali) all'atto della chiamata.</p>	
<p>2. [4]</p>	<p>Si considerino le caratteristiche del linguaggio C:</p> <p>A [F] 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>B [V] Un'area dati allocata tramite malloc può essere utilizzata anche dopo il termine dell'esecuzione della funzione in cui è stata allocata.</p> <p>C [F] Per eseguire una operazione di lettura da file è necessario che il file sia un file di testo.</p> <p>D [F] Lo stack può contenere variabili allocate dinamicamente (tramite la malloc).</p> <p>E [F] Data una variabile <i>i</i> di tipo <code>int</code>, l'istruzione <code>fread(&i, sizeof(int), 1, stdin);</code> può essere utilizzata per leggere un dato intero da tastiera (ad esempio, inserendo da tastiera 19, la variabile <i>i</i> assume valore 19).</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]); }</pre> <p>A [F] Il ciclo al punto 4 stampa i seguenti valori (in linee separate): 2,1,3,3,4,5,5,7,6.</p> <p>B [V] Al punto 1 la variabile <i>V</i> non è visibile.</p> <p>C [F] Al punto 3 la variabile <i>i</i> ha valore 0.</p> <p>D [V] La variabile <i>a</i> è una variabile automatica.</p> <p>E [V] L'istruzione prima del punto 1 (<code>if(*a<*(a+b))</code>) viene eseguita 3 volte.</p>