



Note allo svolgimento. Ti sono stati consegnati:

- a — il *testo del compito*;
- 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;
3. compilare la scheda delle risposte, indicando sulla stessa con una croce × le risposte ritenute corrette, (vedi foglio di istruzioni). A questo proposito, è importante sapere che:
 - ogni errore determinerà un **punteggio negativo**;
 - se a un quesito non si fornisce alcuna risposta, si otterrà per quel quesito il **punteggio 0**;
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).

È severamente proibito consultare libri, appunti, manuali, o strumenti elettronici (computer, cellulari, palmari ecc.).

La prova dura 1 ora.

<p>1. [3.5]</p>	<p>Si consideri l'architettura di un Personal Computer.</p> <p>A La memoria secondaria e' volatile.</p> <p>B L'accesso ai registri della CPU e' piu' lento rispetto all'accesso alla cache di secondo livello.</p> <p>C La memoria centrale contiene istruzioni del programma correntemente in esecuzione.</p> <p>D La ALU esercita il controllo sui trasferimenti tra CPU e memoria.</p> <p>E Il bus consente lo scambio di informazioni tra dispositivi e memoria.</p>	<p>2. [3.5]</p>	<p>Linguaggi di programmazione.</p> <p>A Le istruzioni di un linguaggio macchina sono espresse mediante una sequenza di bit.</p> <p>B Lo sviluppo di programmi in un linguaggio di alto livello puo' non richiedere l'uso di un traduttore.</p> <p>C Un compilatore per il linguaggio C e' un programma che e' indipendente dall'architettura HW/SW del computer utilizzato.</p> <p>D Il compilatore puo' rilevare errori sintattici nel programma sorgente da tradurre.</p> <p>E L'esecuzione di un programma interpretato e' piu' veloce dell'esecuzione dello stesso programma compilato.</p>
<p>3. [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> main() { int x=0; float y=x; char z='z'; if (!y) if(++x) z='a'; else z='b'; else z=x; /* punto 1 */ y=(x+=2, x--); /*istruzione 1*/ /* punto 2 */ { float A; A=(z>'a'? --y: y/x); /* punto 3 */ } /*punto 4*/ }</pre> <p>A Immediatamente dopo l'istruzione if (punto 1), la variabile z assume il valore 'b'.</p> <p>B Al punto 2 le variabili x e y hanno lo stesso valore.</p> <p>C Al punto 2 la variabile x ha valore 2.</p> <p>D Al punto 3 la variabile A ha valore 1.</p> <p>E Al punto 4 la variabile A non e' visibile.</p>	<p>4. [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> main() { int x; float y; char z, w; x=4; y=-2; z='A'; w='B'; x=(x%(int)y ?++x: w-z); /* istruzione 1 */ y=x++/y; /* istruzione 2 */ w=z+=1; /* istruzione 3 */ }</pre> <p>A Il programma puo' essere eseguito correttamente.</p> <p>B Immediatamente dopo l'esecuzione dell'istruzione 1, la variabile x ha il valore 65.</p> <p>C L'operatore di casting nell'istruzione 1 non e' necessario.</p> <p>D Immediatamente dopo l'esecuzione dell'istruzione 2, la variabile y ha il valore -1.</p> <p>E Immediatamente dopo l'esecuzione dell'istruzione 2, la variabile x ha il valore 2.</p>

<p>5. [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> main() { int V[4]={2, 3, 4, 7}; int i, x, y, z; for(i=3; i>=0; i--)//*istruzione 1*/ switch(V[i]%3) /*istruzione 2*/ { case 1: V[i]--; case 2: x=V[i]; break; case 3: z=0; break; default: y=++x; } z=V[0]+V[3] - y; /* istruzione 3*/ }</pre> <p>A Immediatamente dopo il ciclo for, la variabile z ha sempre il valore 0. B Immediatamente dopo il ciclo for, la variabile y ha il valore 4. C I rami dell'istruzione switch vengono eseguiti in modo mutuamente esclusivo. D L'esecuzione del ciclo for provoca 3 iterazioni. E Immediatamente dopo l'istruzione 3, la variabile z ha il valore 0.</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> #define N 8 main() { typedef float new[N]; new A,B={0,0.5,1,1.5,2.0,2.5,3.0,3.5}; int i; A[0]=0; for (i=1; i<N; i+=i) A[i]=B[i-1]/i; /* punto 1 */ if (A[0]?1:A[0]) printf("Ciao!\n"); else { i=A[0]; printf("%d\n", ++i); } /* punto 2*/ }</pre> <p>6. [4]</p> <p>A Il programma stampa "Ciao". B Al punto 1: il valore di A[6] e' 0.416. C Al punto 1: il valore di A[2] e' 0.25. D Al punto 1: a tutti gli elementi di A e' stato assegnato un valore. E Nel programma vi sono tipi non primitivi.</p>
<p>7. [3.5]</p>	<p>Si considerino i vettori nel linguaggio C:</p> <p>A Gli elementi di un vettore sono tutti di uno ed un solo tipo. B I valori degli elementi di un vettore di interi sono sempre ordinati in ordine crescente. C Nella definizione di un vettore, la dimensione puo' essere espressa mediante il nome di una variabile. D L'indice di un elemento puo' essere una variabile di tipo char. E L'indice di un elemento puo' essere una variabile di tipo float.</p>	<p>Si considerino i tipi di dato scalari primitivi nel linguaggio C:</p> <p>A E' possibile eseguire una somma tra una variabile double e una variabile char B E' possibile applicare l'operatore % (modulo, o resto) a una variabile double e una variabile int. 8. [3.5] C L'applicazione della regola di conversione implicita di tipo produce sempre una espressione in cui gli operandi sono omogenei in tipo. D Dopo l'applicazione della regola di conversione implicita di tipo, ogni espressione e' sempre eseguibile. E La differenza tra un float e un int produce un valore di tipo int.</p>
<p>9. [3.5]</p>	<p>Si considerino le istruzioni del linguaggio C:</p> <p>A Ogni istruzione switch e' sempre un'alternativa multipla. B Ogni istruzione while e' sempre esprimibile mediante un'istruzione for. C Ogni istruzione if e' sempre esprimibile mediante una espressione condizionale. D In un'istruzione switch: il blocco di istruzioni associato all'etichetta default viene sempre eseguito. E In un'istruzione for l'istruzione da ripetere viene sempre eseguita almeno una volta.</p>	