



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 (**G**) 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. A L'IR contiene il programma correntemente in esecuzione. B La ALU esegue operazioni matematiche. C Il bus consente il trasferimento di istruzioni dalla memoria alla CPU. D Il bus consente esclusivamente la trasmissione unidirezionale tra memoria e CPU. E Il clock comunica l'ora corrente ai dispositivi che la richiedono.</p>	<p>2. [3.5]</p> <p>Linguaggi di programmazione. A Un compilatore per un linguaggio e' indipendente dall'architettura HW/SW del computer utilizzato. B Il compilatore serve solo per rilevare gli errori sintattici in un programma. C L'esecuzione di un programma interpretato e' tipicamente piu' veloce dell'esecuzione dello stesso programma compilato. D Un programma scritto in linguaggio assembler necessita di un traduttore. E Un programma compilato e linkato e' sicuramente privo di errori.</p>
<p>3. [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> main() { int x=2; float y=x--; char z='y'; if (--y) if(--x) z='a'; else z='b'; else z=x; /* punto 1 */ y=(x+=3, x--); /*istruzione 1*/ /* punto 2 */ { float A; A=(z>'a'? --y: y/x); /* punto 3*/ } /*punto 4*/ }</pre> <p>A Al punto 2 la variabile A vale 0. B Al punto 3 la variabile A ha valore 2.0. C Al punto 4 la variabile A non e' visibile. D Al punto 1 la variabile x ha valore 1. E L'istruzione 1 e' equivalente alla sequenza di istruzioni: x=x+3; x--; y=x;</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='B'; w='C'; x=(x%(int)y ?++x: w-z); /* istruzione 1 */ y=x++/y; /* istruzione 2 */ w=z+1; w-=z; /* istruzione 3 */ }</pre> <p>A L'operatore di casting nell'istruzione 1 non e' necessario. B Immediatamente dopo l'esecuzione dell'istruzione 2, la variabile y ha il valore -1. C Immediatamente dopo l'esecuzione dell'istruzione 2, la variabile x ha il valore 2. D Immediatamente dopo l'esecuzione dell'istruzione 3, le variabili z e w hanno lo stesso valore. E Immediatamente dopo l'esecuzione dell'istruzione 3, la variabile z ha valore 'C'.</p>

<p>5. [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> main() {int V[4]={0, 3, 4, 8}; int i, x, y=0, z; for(i=3; i>=0; i--)/ *istruzione 1*/ switch(V[i]%3-1) /*istruzione 2*/ {case 0: V[i]--; case 1: x=V[i]; break; case 2: z=0; break; default: y=-x; } z=V[0]+V[2] - y; /* istruzione 3*/ }</pre> <p>A Il blocco di istruzioni corrispondente al case 1: viene eseguito 2 volte. B L'esecuzione del ciclo for provoca 3 iterazioni. C Immediatamente dopo l'istruzione 3, la variabile z ha il valore 2. D Immediatamente dopo l'istruzione 3, la variabile x ha il valore 1. E Al termine dell'esecuzione, il valore di uno solo tra gli elementi di V risulta modificato rispetto al suo valore iniziale.</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> #define N 8 main() {typedef float f[N]; typedef char c[N]; f A={0,0.5,1,1.5,2.0,2.5,3.0,3.5}; c B={'C','i','a','o',0,1,2,3}; int i; A[N-1]=0; for (i=1; i<N; i+=i) A[i]=B[N-i]/i; /* punto 1 */ if (A[--i]?1:A[0]) printf("Ciao!\n"); else {i=A[0]; for (;i<5;) printf("%c", B[i++]); } /* punto 2 */ }</pre> <p>A Al punto 2: il valore di A[2] e' 2.0. B Al punto 1: tutti gli elementi di A tranne gli estremi sono reali strettamente positivi. C Nel programma vi sono tipi non primitivi. D Al punto 2: la variabile i ha valore 5. E Il programma contiene un errore di sintassi.</p>
<p>7. [3.5]</p>	<p>Si considerino i vettori nel linguaggio C:</p> <p>A Nella definizione di un vettore, la dimensione puo' essere espressa mediante il nome di una variabile. B L'indice di un elemento puo' essere una variabile di tipo char. C L'indice di un elemento puo' essere una variabile di tipo float. D L'indice di un elemento di un vettore deve essere minore della sua dimensione logica. E Non e' possibile applicare l'operatore di assegnamento a variabili di tipo vettore.</p>	<p>Si considerino i tipi di dato scalari primitivi nel linguaggio C:</p> <p>A L'applicazione della regola di conversione implicita di tipo e' sempre possibile. B La coercizione puo' trasformare un float in un char. C La sottrazione tra un float e un char produce un valore di tipo float. D Il criterio di equivalenza strutturale puo' consentire l'assegnamento del valore di una variabile di tipo non primitivo a una variabile di tipo scalare primitivo. E L'<i>overloading</i> degli operatori permette di elaborare con lo stesso operatore tipi di dati altrimenti incompatibili .</p> <p>8. [3.5]</p>
<p>9. [3.5]</p>	<p>Si considerino le istruzioni del linguaggio C:</p> <p>A Ogni istruzione <code>if</code> e' sempre esprimibile mediante un assegnamento. B Nell'istruzione <code>switch</code> il blocco di istruzioni associato all'etichetta <code>default</code> viene sempre eseguito. C Nell'istruzione <code>for</code> il blocco di istruzioni puo' non essere mai eseguito. D Ogni istruzione <code>switch</code> e' sempre esprimibile mediante altre istruzioni. E Nell'istruzione <code>for</code> l'espressione di modifica viene eseguita nel momento in cui la condizione e' falsa.</p>	