



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 (**F**) 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 Il registro Flag (o PSW) contiene informazioni riguardanti l'ultima operazione eseguita dalla ALU. B Ad ogni ciclo il registro IR contiene l'istruzione da eseguire nello stesso ciclo. C La memoria centrale puo' contenere i dati del programma correntemente in esecuzione. D L'accesso alla cache di secondo livello e' piu' veloce rispetto all'accesso alla RAM. E La memoria di massa e' volatile.</p>	<p>2. [3.5]</p>	<p>Linguaggi di programmazione. A Un programma espresso in linguaggio macchina e' una sequenza di bit. B Un compilatore traduce un programma scritto con un linguaggio di alto livello in uno scritto con un linguaggio a basso livello, ma non viceversa. C Lo sviluppo di programmi in linguaggio assembler necessita dell'uso di un traduttore. D Un compilatore per il linguaggio C e' un programma che e' indipendente dall'architettura HW/SW del computer utilizzato. E L'esecuzione di un programma compilato è tipicamente più veloce dell'esecuzione dello stesso programma interpretato.</p>
<p>3. [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> main() { int A=2; float B=A--; char C=A; if (--B) if(--A) C='B'; else C='A'; else C=A; /* punto 1 */ B=(A+=3, A--); /*istruzione 1*/ /* punto 2 */ { float num; num=(C>'B'? B--: B/A); /* punto 3 */ } /*punto 4*/ }</pre> <p>A Al punto 1 la variabile C assume il valore 'B'. B L'istruzione 1 e' equivalente alla sequenza di istruzioni: A=A+3; B=A; --A; C Al punto 2 la variabile A ha valore 2. D Al punto 2 le variabili A e B hanno valore diverso. E Al punto 3 la variabile num ha valore 1.5.</p>	<p>4. [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> main() {int A; float B; char C, ch; A=4; B=-2; C='C'; ch='D'; A=(A%(int)B ?++A: ch-C); /* istruzione 1 */ B=A++/B; /* istruzione 2 */ ch=C+=1; ch-=C; /* istruzione 3 */ }</pre> <p>A Immediatamente dopo l'esecuzione dell'istruzione 2, la variabile A ha il valore 2. B Immediatamente dopo l'esecuzione dell'istruzione 3, la variabile C ha valore 'C'. C L'operatore di casting nell'istruzione 1 e' ininfluente. D Il programma puo' generare errori durante l'esecuzione. E Immediatamente dopo l'esecuzione dell'istruzione 2, la variabile B ha il valore -0.5.</p>

<p>5. [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> main() {int x[4]={0, 3, 4, 8}; int j, A, B=0, C; for(j=3; j>=0; j--) /*istruzione 1*/ switch(x[j]%3-1) /*istruzione 2*/ {case 0: x[j]--; case 1: A=x[j]; break; case 2: C=0; break; default: B--A; } C=x[0]+x[2] - B; /* istruzione 3*/ }</pre> <p>A Immediatamente dopo il ciclo for, la variabile B ha il valore 1. B L'esecuzione del ciclo for provoca 4 iterazioni. C Immediatamente dopo il ciclo for, la variabile C vale 0. D Il blocco di istruzioni corrispondente al case 1: viene eseguito 2 volte. E Immediatamente dopo l'istruzione 3, la variabile C ha il valore 3.</p>	<p>Si consideri il seguente programma C:</p> <pre>#include <stdio.h> #define N 8 main() {typedef float V1[N]; typedef char V2[N]; V1 x={0,0.5,1,1.5,2.0,2.5,3.0,3.5}; V2 y={'C','i','a','o',0,1,2,3}; int i; x[N-1]=0; for (i=1; i<N; i+=i) x[i]=y[N-i]/i; /* punto 1 */ if (x[--i]?1:x[0]) printf("Ciao!\n"); else {i=x[0]; for (;i<=4;) printf("%c", y[i++]); } /* punto 2*/ }</pre> <p>A Al punto 1: il valore di x[1] e' 3. B Nel programma non vengono usati tipi di dato non primitivi. C Il programma contiene un errore di sintassi. D Al punto 2: x[i-1] ha valore 0.0. E Al punto 1: il valore di x[4] e' 1.5.</p>
<p>7. [3.5]</p>	<p>Si considerino i vettori nel linguaggio C:</p> <p>A E' possibile copiare gli elementi di un vettore in un altro vettore. B Gli elementi di uno stesso vettore possono essere di tipo diverso. C E' possibile stampare il contenuto di un vettore di 5 interi con una sola printf. D E' possibile ordinare un vettore di interi in ordine crescente. E Nella definizione di un vettore, la dimensione puo' essere espressa mediante una espressione matematica.</p>	<p>Si considerino i tipi di dato scalari primitivi nel linguaggio C:</p> <p>A L'<i>overloading</i> degli operatori permette di denotare operatori diversi con lo stesso simbolo. B E' possibile applicare l'operatore % (modulo, o resto) a una variabile float e una variabile int. C La sottrazione tra un float e un char produce un errore in compilazione. D Nel linguaggio C il vettore e' un tipo scalare primitivo. E L'applicazione della regola di conversione implicita produce una espressione in cui tutti gli operandi assumono uno stesso tipo.</p>
<p>9. [3.5]</p>	<p>Si considerino le istruzioni del linguaggio C:</p> <p>A Nell'istruzione <code>do..while</code> se la condizione non e' verificata, si esce dal ciclo. B Nell'istruzione <code>switch</code> puo' essere eseguito al più un solo blocco di istruzioni. C Ogni istruzione <code>switch</code> e' sempre esprimibile mediante uno o piu' <code>if</code> (eventualmente annidati). D Nell'istruzione <code>for</code> l'espressione di modifica viene sempre eseguita alla fine di ogni iterazione. E Ogni istruzione <code>for</code> e' sempre esprimibile mediante <code>while</code>.</p>	