

<p><b>1.</b> [3.5]</p>	<p>Si consideri l'architettura di un Personal Computer.  <b>A [V]</b> La memoria centrale puo' contenere i dati del programma correntemente in esecuzione.  <b>B [V]</b> L'accesso alla cache di secondo livello e' piu' veloce rispetto all'accesso alla RAM.  <b>C [F]</b> La memoria di massa e' volatile.  <b>D [V]</b> La Control Unit (CU) esercita il controllo sui trasferimenti tra CPU e memoria.  <b>E [V]</b> Il bus puo' essere utilizzato per il trasferimento di dati e istruzioni tra memoria e CPU.</p>	<p><b>2.</b> [3.5]</p>	<p>Linguaggi di programmazione.  <b>A [V]</b> Lo sviluppo di programmi in linguaggio assembler necessita dell'uso di un traduttore.  <b>B [F]</b> Un compilatore per il linguaggio C e' un programma che e' indipendente dall'architettura HW/SW del computer utilizzato.  <b>C [V]</b> L'esecuzione di un programma compilato e' tipicamente più veloce dell'esecuzione dello stesso programma interpretato.  <b>D [F]</b> Un programma scritto in linguaggio assembler non necessita di traduzione.  <b>E [F]</b> Il debugger serve per trovare più facilmente errori sintattici nei programmi.</p>
<p><b>3.</b> [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include &lt;stdio.h&gt;  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&gt;'B'? B--: B/A);     /* punto 3*/   }   /*punto 4*/ }</pre> <p><b>A [V]</b> Al punto 2 la variabile A ha valore 2.  <b>B [V]</b> Al punto 2 le variabili A e B hanno valore diverso.  <b>C [V]</b> Al punto 3 la variabile num ha valore 1.5.  <b>D [F]</b> Al punto 4 la variabile num e' visibile.  <b>E [F]</b> L'istruzione 1 effettua 2 assegnamenti.</p>	<p><b>4.</b> [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include &lt;stdio.h&gt;  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><b>A [F]</b> L'operatore di casting nell'istruzione 1 e' ininfluenza.  <b>B [F]</b> Il programma puo' generare errori durante l'esecuzione.  <b>C [V]</b> Immediatamente dopo l'esecuzione dell'istruzione 2, la variabile B ha il valore -0.5.  <b>D [V]</b> Immediatamente dopo l'esecuzione dell'istruzione 3, la variabile ch ha valore '0'.  <b>E [F]</b> Immediatamente dopo l'esecuzione dell'istruzione 1, la variabile A ha il valore corrispondente al carattere ASCII 'A'.</p>

<p>5. [4]</p>	<p>Si consideri il seguente programma C:</p> <pre>#include &lt;stdio.h&gt;  main() {int x[4]={0, 3, 4, 8};  int j, A, B=0, C;  for(j=3; j&gt;=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 [F] Immediatamente dopo il ciclo for, la variabile C vale 0.  B [V] Il blocco di istruzioni corrispondente al case 1: viene eseguito 2 volte.  C [F] Immediatamente dopo l'istruzione 3, la variabile C ha il valore 3.  D [F] Al termine dell'esecuzione, il valore di due elementi di V risultano modificati rispetto ai loro valori iniziali.  E [F] Al termine dell'esecuzione, il valore di V[3] e' uguale a 7.</p>	<p>Si consideri il seguente programma C:</p> <pre>#include &lt;stdio.h&gt; #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&lt;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&lt;=4;) printf("%c", y[i++]);   }  /* punto 2*/ }</pre> <p>6. [4]</p> <p>A [F] Il programma contiene un errore di sintassi.  B [V] Al punto 2: x[i-1] ha valore 0.0.  C [F] Al punto 1: il valore di x[4] e' 1.5.  D [F] Al punto 1: almeno un elemento di y e' stato modificato.  E [V] Al punto 2: la variabile i ha valore N-3.</p>
<p>7. [3.5]</p>	<p>Si considerino i vettori nel linguaggio C:</p> <p>A [V] E' possibile stampare il contenuto di un vettore di 5 interi con una sola printf.  B [V] E' possibile ordinare un vettore di interi in ordine crescente.  C [V] Nella definizione di un vettore, la dimensione puo' essere espressa mediante una espressione matematica.  D [V] L'indice di un elemento di un vettore deve essere minore della sua dimensione fisica.  E [V] L'indice di un elemento puo' essere una variabile di tipo char.</p>	<p>Si considerino i tipi di dato scalari primitivi nel linguaggio C:</p> <p>A [F] La sottrazione tra un float e un char produce un errore in compilazione.  B [F] Nel linguaggio C il vettore e' un tipo scalare primitivo.  C [F] L'applicazione della regola di conversione implicita produce una espressione in cui tutti gli operandi assumono uno stesso tipo.  D [F] La conversione esplicita puo' trasformare un dato scalare in uno strutturato.  E [V] 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.</p> <p>8. [3.5]</p>
<p>9. [3.5]</p>	<p>Si considerino le istruzioni del linguaggio C:</p> <p>A [V] Ogni istruzione switch e' sempre esprimibile mediante uno o piu' if (eventualmente annidati).  B [V] Nell'istruzione for l'espressione di modifica viene sempre eseguita alla fine di ogni iterazione.  C [V] Ogni istruzione for e' sempre esprimibile mediante while.  D [F] Ogni istruzione if e' esprimibile mediante un unico ciclo while.  E [F] Non e' possibile usare un if all'interno di uno switch.</p>	