



Nome

Cognome

Matricola

Domande

Domanda D1	<p>Si consideri il seguente metodo:</p> <pre>void metodo1_A (int vett[], int vettSize, int x) { for(int i = 0; i < vettSize; i++) if(vett[i] % x != 0) vett[i] = x; else vett[i] = 1 - vett[i] - x; }</pre> <p>Indicare il contenuto della variabile vett dopo l'esecuzione del seguente frammento di codice:</p> <pre>int vett[] = {9,8,7,6,5,4}; metodo1_A(vett,6,2);</pre>
Risposta R1	<ol style="list-style-type: none"> 1. vett = {-10,2,-8,2,-6,2} 2. vett = {-9,-8,-7,-6,-5,-4} 3. vett = {2,-9,2,-7,2,-5}

Domanda D2	<p>Si consideri il seguente metodo:</p> <pre>int metodo2_A (int vett[], int vettSize, int x) { int i = 0, j = 0; while (i < vettSize) if(vett[i++] % x == 0) j++; else j--; return j; }</pre> <p>Indicare il contenuto della variabile res dopo l'esecuzione del seguente frammento di codice:</p> <pre>int vett[] = {1,1,4,2,3,3}; int res = metodo2_A(vett,6,4);</pre>
Risposta R2	<ol style="list-style-type: none"> 1. res = 2 2. res = -4 3. res = -5

Domanda D3	<p>Si considerino le seguenti istruzioni:</p> <pre>1 const int y_A = 5; 2 int *x1_A = new int [5]; 3 int *x2_A = new int [y_A]; 4 int x3_A[] = {3,5,1,6}; 5 int x4_A[y_A] = new *int[y_A]; 6 int *x5_A = *x3_A; 7 int x6_A[4];</pre> <p>Indicare, se presenti, la/e linea/e di codice contenente/i errore/i</p>
Risposta R3	Linea 5 e linea 6

Domanda D4	<p>Si consideri il seguente metodo:</p> <pre>int metodo3_A (int vett[], int vettSize) { if(vettSize > 0) return metodo3_A(vett,vettSize - 1) + vett[vettSize - 1]; return 0; }</pre> <p>Quale sarà il contenuto della variabile res dopo l'esecuzione del seguente frammento di codice:</p> <pre>int vett[] = {2,1,1,1,7,6}; int res = metodo3_A (vett,6);</pre>
Risposta R4	<ol style="list-style-type: none"> 1. res = 16 2. res = 18 3. res = 0

Domanda D5	<p>Si consideri la seguente dichiarazione:</p> <pre>1 class X_A { 2 public: 3 X_A(int size = 0); 4 X_A(const bool* b); 5 X_A(const X_A& x); 6 ~X_A(); 7 const X_A& operator=(const X_A& x); 8 bool operator==(const X_A& x) const; 9 private: 10 bool* b; int size; 11 };</pre> <p>Indicare, se presenti, la/e linea/e di codice corrispondente/i al prototipo del costruttore di copia</p>
Risposta R5	Linea 5

Domanda D6	<p>In riferimento alla definizione di classe della domanda D5 quale sarà il codice del distruttore:</p> <pre>X_A::~~X_A() { /*CODICE?*/ }</pre> <p>se i vari costruttori, tra le altre cose, eseguono la seguente istruzione:</p> <pre>b = new bool[size];</pre>
Risposta R6	<ol style="list-style-type: none"> 1. delete []b; 2. delete b; delete size; 3. delete b[];

Domanda D7	<p>In riferimento alla classe della domanda D5 e dato il seguente frammento di codice:</p> <pre>X_A xa1(); X_A xa2(5) xa1 = X_A(xa2);</pre> <p>indicare l'esatta sequenza dei metodi invocati facendo riferimento alla numerazione delle relative linee di codice</p>
Risposta R7	<ol style="list-style-type: none"> 1. [3] → [3] → [5] → [7] 2. [3] → [4] → [5] → [5] 3. [6] → [3] → [7]

Domanda D8	In riferimento al metodo <code>const X_A& operator=(const X_A& x);</code> della classe <code>X_A</code> (domanda D5 , linea 7), quale frammento di codice deve essere necessariamente eseguito nel caso in cui i due oggetti hanno dimensioni differenti?
Risposta R8	1. <code>this->size = x.size; b = new bool[size]; delete []b;</code> 2. <code>delete []b; size = x.size; this->b = new bool[size];</code> 3. <code>delete []b; b = new bool[size]; this->size = x.size;</code>

Domanda D9	In riferimento alla definizione di classe della domanda D5 e dato il seguente frammento di codice: <pre> 1 X_A x1a(3); 2 X_A x2a = x1a; 3 x1a = x2a; 4 X_A x3a(x1a); 5 x1a = X_A(x3a); </pre> Indicare, se presenti, la/e linea/e di codice dove viene invocato il costruttore di copia
Risposta R9	Linea 2, linea 4 e linea 5

Domanda D10	Si considerino le seguenti dichiarazioni: <pre> 1 class X_A { 2 public: int pbc_xa; 3 protected: int prt_xa; 4 private: int prv_xa; 5 }; 6 class Y_A: private X_A { 7 public: int pbc_ya() {return pbc_xa;} 8 protected: int prt_ya() {return pbc_xa + prt_xa; } 9 private: int prv_ya() {return pbc_xa + prt_xa + prv_xa;} 10 }; 11 class Z_A: protected Y_A { 12 public: 13 int pbc_z() { return pbc_ya(); } 14 int prt_z() { return prt_ya(); } 15 int prv_z() { return prv_ya(); } 16 }; </pre> Indicare, se presenti, la/e linea/e di codice contenente/i errore/i
Risposta R10	Linea 9 e linea 15