
Test 1

1. Cosa fa la seguente funzione `esegui()` se `testa` è un puntatore alla testa di una lista concatenata?

C++

```
void esegui(Nodo* testa) {  
    if (testa == nullptr) return;  
    cout << testa->valore << endl;  
    esegui(testa->succ);  
}
```

- a) Stampa il contenuto della lista in ordine inverso.
 - b) Stampa il contenuto della lista dalla testa alla coda.
 - c) Stampa solo il primo e l'ultimo elemento della lista.
 - d) Il codice causa un ciclo infinito.
2. Qual è la politica di accesso ai dati implementata da una Coda (Queue)?
- a) LIFO (Last-In, First-Out)
 - b) FIFO (First-In, First-Out)
 - c) Accesso casuale
 - d) Nessuna delle precedenti
3. Qual è la complessità temporale nel caso peggiore per la ricerca di un elemento in un Albero Binario di Ricerca (BST) non bilanciato di n nodi?
- a) $O(1)$
 - b) $O(\log n)$
 - c) $O(n)$
 - d) $O(n^2)$
4. In C++, quale operatore viene utilizzato per deallocare la memoria allocata dinamicamente per un singolo oggetto?
- a) `new`
 - b) `malloc`
 - c) `delete`
 - d) `remove`
5. Considera una classe `Animale`. Se `class Cane : public Animale`, quale delle seguenti affermazioni è vera?
- a) `Animale` è una classe derivata e `Cane` è una classe base.
 - b) `Cane` eredita solo i membri `public` di `Animale`.
 - c) `Cane` è una classe derivata e `Animale` è una classe base.
 - d) `Cane` non può accedere ai membri `protected` di `Animale`.

6. **Quale frammento di codice inserisce correttamente un nuovoNodo in testa a una lista concatenata puntata da testa?**

- a) nuovoNodo->succ = testa; testa = nuovoNodo;
- b) testa->succ = nuovoNodo;
- c) nuovoNodo->succ = testa;
- d) testa = nuovoNodo; nuovoNodo->succ = testa;

7. **Qual è la complessità della seguente funzione?**

C++

```
void func(int n) {  
    for (int i = 1; i < n; i = i * 2) {  
        cout << "Test" << endl;  
    }  
}
```

- a) $O(n)$
- b) $O(n^2)$
- c) $O(\log n)$
- d) $O(n \log n)$

8. **Una funzione definita come `virtual void muovi() = 0;` all'interno di una classe la rende:**

- a) Una classe template
- b) Una classe astratta
- c) Una classe friend
- d) Una classe statica

9. **Cosa rappresenta un "riferimento" (&) in C++?**

- a) Un puntatore al valore di una variabile.
- b) Un alias, ovvero un altro nome per una variabile esistente.
- c) Una copia del valore di una variabile.
- d) L'indirizzo di memoria di una variabile.

10. **Supponendo di avere un puntatore alla testa e alla coda di una lista doppiamente concatenata, quale operazione ha un costo $O(1)$?**

- a) Ricerca di un elemento per valore.
- b) Inserimento di un elemento in una posizione specifica (diversa da testa e coda).
- c) Eliminazione dell'ultimo elemento.
- d) Invertire la lista.

11. **Come si apre un file chiamato "dati.txt" in modalità scrittura in C++?**

- a) `ifstream file("dati.txt");`
- b) `ofstream file("dati.txt");`
- c) `fstream file("dati.txt", "w");`
- d) `open("dati.txt", "write");`

12. **A cosa serve la parola chiave `this` all'interno di un metodo di una classe?**

- a) A creare un nuovo oggetto della classe.
- b) A distruggere l'oggetto corrente.
- c) A puntare all'istanza dell'oggetto che ha invocato il metodo.
- d) A definire un metodo statico.

13. **Qual è l'output del seguente codice C++?**

C++

```
std::string testo = "programma";  
cout << testo.substr(4, 3);
```

a) gram b) ram c) ogra d) amma

14. **Quale di queste non è una caratteristica del polimorfismo in C++?**

- a) Overloading di funzioni.
- b) Funzioni virtuali.
- c) Ereditarietà multipla.
- d) Overriding di metodi.

15. **In una lista concatenata semplice, per eliminare un nodo di cui si ha il puntatore, quale informazione aggiuntiva è necessaria?**

- a) Il puntatore alla coda della lista.
- b) Il puntatore al nodo precedente.
- c) La lunghezza totale della lista.
- d) Nessuna, il puntatore al nodo è sufficiente.

16. **Qual è la complessità della funzione seguente?**

C++

```
int func(int n) {  
    if (n <= 1) return 1;  
    return func(n - 1) + func(n - 1);  
}
```

a) $O(n)$ b) $O(\log n)$ c) $O(n^2)$ d) $O(2n)$

17. **Cosa si intende per "template" in C++?**

- a) Una classe con solo metodi statici.
- b) Un modello per creare classi o funzioni che operano su tipi di dato generici.
- c) Un file di intestazione (.h).
- d) Una libreria standard di C++.

18. **Cosa accade se si tenta di accedere a un elemento di un array oltre il suo limite?**

- a) Il programma genera sempre un errore di compilazione.
- b) Viene sollevata un'eccezione `std::out_of_range`.
- c) Il comportamento è indefinito (undefined behavior).
- d) L'array viene automaticamente ridimensionato.

19. **Quale traversata di un Albero Binario di Ricerca (BST) produce gli elementi in ordine non decrescente?**

- a) Pre-ordine
- b) Post-ordine
- c) In-ordine
- d) Level-ordine

20. **L'implementazione di una Pila (Stack) tramite un array statico:**

- a) Non è possibile.
- b) Rende le operazioni di push e pop più complesse.
- c) Limita la dimensione massima della pila.
- d) Trasforma la pila in una coda.

1. **Quale delle seguenti strutture dati segue una politica LIFO (Last-In, First-Out)?**
 - a) Coda (Queue)
 - b) Pila (Stack)
 - c) Lista doppiamente concatenata
 - d) Albero binario di ricerca
2. **Qual è la complessità della seguente funzione?**

C++

```
void calcola(int n) {  
    int k = 0;  
    for (int i = 0; i < n; i++) {  
        for (int j = i; j < n; j++) {  
            k++;  
        }  
    }  
}
```

- a) $O(n \log n)$
 - b) $O(n)$
 - c) $O(n^2)$
 - d) $O(\log n)$
3. **In C++, come si alloca dinamicamente un array di 100 interi?**
 - a) `int* arr = new int(100);`
 - b) `int arr[100];`
 - c) `int* arr = new int[100];`
 - d) `int* arr = alloc int[100];`
 4. **Una classe B eredita privatamente da una classe A (`class B : private A`). Come vengono visti i membri `public` e `protected` di A all'interno di B?**
 - a) Diventano `public` in B.
 - b) Diventano `protected` in B.
 - c) Diventano `private` in B.
 - d) Non sono accessibili da B.
 5. **Cosa fa la funzione `func()` se p è la testa di una lista concatenata?**

C++

```
int func(Nodo* p) {  
    return (p == nullptr || p->succ == nullptr || (p->valore >= p->succ->valore && func(p->succ)));  
}
```

- a) Restituisce 1 se e solo se la lista è ordinata in modo non decrescente.
 - b) Restituisce 1 se e solo se la lista è ordinata in modo non crescente.
 - c) Restituisce 1 se e solo se la lista contiene elementi duplicati.
 - d) Restituisce sempre 0.
6. **Qual è il vantaggio principale di una lista doppiamente concatenata rispetto a una semplice?**
 - a) Occupa meno memoria.
 - b) Permette la traversata in entrambe le direzioni (avanti e indietro).

- c) L'accesso a un elemento casuale è più veloce.
- d) Le operazioni di inserimento in testa sono più efficienti.

7. Cosa fa il **break**; nel ciclo interno di questo codice?

C++

```
for(int i = 1; i <= n; i++) {  
    for(int j = 1; j <= n; j++) {  
        cout << "X";  
        break;  
    }  
}
```

- a) Termina l'esecuzione di entrambi i cicli.
 - b) Termina solo l'esecuzione del ciclo interno per l'iterazione corrente.
 - c) Salta l'iterazione corrente del ciclo esterno.
 - d) Termina il programma.
8. Quale dei seguenti non è un tipo di ereditarietà in C++?
- a) public
 - b) protected
 - c) private
 - d) friend
9. Cosa si intende per "dangling pointer" (puntatore penzolante)?
- a) Un puntatore che punta all'indirizzo `nullptr`.
 - b) Un puntatore che punta a un'area di memoria che è già stata deallocata.
 - c) Un puntatore non inizializzato.
 - d) Un puntatore a un puntatore.
10. Quale frammento di codice elimina correttamente il nodo puntato da **x** in una lista doppiamente concatenata (assumendo che **x** non sia né testa né coda)?
- a) `X->prec->succ = X->succ; X->succ->prec = X->prec; delete X;`
 - b) `X->prec = X->succ; delete X;`
 - c) `X->succ->prec = X->prec; delete X;`
 - d) `delete X;`
11. Qual è la complessità media per inserire un elemento in un Albero Binario di Ricerca (BST) bilanciato?
- a) $O(n)$
 - b) $O(1)$
 - c) $O(n \log n)$
 - d) $O(\log n)$
12. Come si definisce un costruttore di copia per una classe **MiaClasse**?
- a) `MiaClasse(MiaClasse other);`
 - b) `MiaClasse(const MiaClasse &other);`
 - c) `void MiaClasse(MiaClasse other);`
 - d) `MiaClasse();`
13. Se **s1** e **s2** sono `std::string`, quale operatore si usa per concatenarle?
- a) *
 - b) &
 - c) +
 - d) .
14. Un metodo di classe dichiarato con la parola chiave **static**:
- a) Può essere chiamato solo da oggetti della classe.

- b) Può essere chiamato senza creare un'istanza della classe.
 - c) Non può accedere a nessun membro della classe.
 - d) Deve essere `virtual`.
15. **Per implementare una Coda (Queue) con un array, quali sono gli indici/puntatori tipicamente necessari?**
- a) Solo un puntatore alla cima (top).
 - b) Un puntatore alla testa (head) e uno alla coda (tail).
 - c) Un puntatore all'elemento centrale.
 - d) Nessun puntatore, solo la dimensione.
16. **Qual è il numero minimo di campi necessari per un nodo di una lista doppiamente concatenata?**
- a) 1
 - b) 2
 - c) 3
 - d) 4
17. **Cosa restituisce la funzione `is_open()` di un oggetto `fstream`?**
- a) Il nome del file.
 - b) Un valore booleano che indica se il file è stato aperto con successo.
 - c) La dimensione del file.
 - d) Un puntatore all'inizio del file.
18. **L'operatore `->` in C++ è una scorciatoia per:**
- a) Dereferenziare un puntatore e poi accedere a un suo membro.
 - b) Accedere a un membro di un oggetto riferimento.
 - c) Ottenere l'indirizzo di un membro di un oggetto.
 - d) Confrontare due puntatori.
19. **Qual è l'output del seguente codice?**

C++

```
int n = 16;
while (n > 1) {
    cout << "***";
    n /= 2;
}
```

- a) * b) ** c) *** d) ****
20. **Una funzione template `template <typename T> T max(T a, T b):`**
- a) Funziona solo con tipi numerici.
 - b) È un modello per una funzione che può operare con diversi tipi di dato.
 - c) Deve essere definita per ogni tipo con cui viene usata.
 - d) Non può essere overloaded.

Test 3

1. Quale operazione su una lista concatenata semplice ha una complessità che dipende dalla lunghezza n della lista?
- a) Inserimento di un nuovo elemento in testa.
 - b) Eliminazione del primo elemento.
 - c) Aggiunta di un nuovo elemento in coda (avendo solo il puntatore alla testa).

d) Controllare se la lista è vuota.

2. Qual è la complessità della seguente funzione?

```
void func(int n) {  
    for (int i = 0; i < n * n; i++) {  
        cout << "step";  
    }  
}
```

- a) $O(n)$
- b) $O(n^2)$
- c) $O(\log n)$
- d) $O(n \log n)$

3. Cosa fa un "distruttore" in una classe C++?

- a) Crea e inizializza un oggetto.
- b) Viene chiamato automaticamente quando un oggetto esce dal suo scope o viene deallocato.
- c) Copia un oggetto in un altro.
- d) Rende tutti i membri private.

4. In una traversata "pre-ordine" di un albero binario, quale nodo viene visitato per primo?

- a) Il nodo più a sinistra.
- b) La radice.
- c) Il nodo più a destra.
- d) Un nodo foglia.

5. Qual è la differenza fondamentale tra un puntatore e un riferimento?

- a) Un puntatore può essere nullo, un riferimento no.
- b) Un puntatore memorizza un valore, un riferimento un indirizzo.
- c) Non c'è alcuna differenza, sono sinonimi.
- d) Un riferimento non può essere passato come argomento di una funzione.

6. Come si dealloca la memoria per un array `arr` allocato con `int* arr = new int[50];`?

- a) `delete arr;`
- b) `delete[] arr;`
- c) `free(arr);`
- d) `arr.delete();`

7. Il polimorfismo a tempo di esecuzione (run-time) in C++ è implementato principalmente tramite:

- a) Template
- b) Overloading di funzioni
- c) Funzioni virtuali
- d) Ereditarietà privata

8. Quale operazione è tipicamente associata a una Pila (Stack)?

- a) enqueue
- b) dequeue
- c) pop

d) find

9. L'istruzione `cin >> valore;` serve a:

- a) Scrivere il contenuto di `valore` sullo standard output.
- b) Leggere un dato dallo standard input e memorizzarlo nella variabile `valore`.
- c) Aprire un file in lettura.
- d) Concatenare due stringhe.

10. Qual è l'output del seguente frammento di codice?

```
string s = "esempio";  
s[0] = 'E';  
s.pop_back();  
cout << s;
```

- a) esempio
- b) Esempio
- c) esempi
- d) Esempi

11. Qual è la complessità del seguente ciclo?

```
for (int i = n; i > 0; i--) {  
    for (int j = 1; j < n; j = j * 2) {  
        // operazioni O(1)  
    }  
}
```

- a) $O(n^2)$
- b) $O(\log n)$
- c) $O(n \log n)$
- d) $O(n)$

12. In un Albero Binario di Ricerca, dove si trova il valore massimo?

- a) Nella radice.
- b) Nel nodo più a sinistra del sottoalbero sinistro.
- c) Nel nodo più a destra del sottoalbero destro.
- d) In una qualsiasi foglia.

13. Se `ptr` è un puntatore a un oggetto, quale delle seguenti è equivalente a `(*ptr).membro`?

- a) `&ptr.membro`
- b) `ptr->membro`
- c) `ptr.membro`
- d) `*ptr.membro`

14. Cosa succede se si dimentica di includere il caso base in una funzione ricorsiva?

- a) Il codice non compila.
- b) La funzione restituisce 0 o `nullptr` di default.
- c) Si verifica un ciclo infinito o un errore di stack overflow.
- d) La funzione viene eseguita una sola volta.

15. Perché si usano i template in C++?

- a) Per migliorare le performance del codice.

- b) Per scrivere codice generico riutilizzabile con tipi di dato diversi.
 - c) Per gestire le eccezioni.
 - d) Per obbligare l'uso del polimorfismo.
16. In una lista doppiamente concatenata, il campo `prec` del nodo di testa a cosa punta?
- a) All'ultimo nodo della lista.
 - b) A sé stesso.
 - c) A `nullptr`.
 - d) Al secondo nodo della lista.
17. Cosa si intende per "memory leak" (perdita di memoria)?
- a) Quando si tenta di accedere a memoria non valida.
 - b) Quando la memoria allocata dinamicamente non viene mai deallocata.
 - c) Quando il programma usa più memoria del previsto.
 - d) Un errore hardware della RAM.
18. Se una classe Derivata eredita da una classe Base, il costruttore di Base viene chiamato:
- a) Dopo il costruttore di Derivata.
 - b) Prima del costruttore di Derivata.
 - c) Non viene chiamato affatto.
 - d) Solo se richiamato esplicitamente con `Base::Base()`.
19. Quale delle seguenti affermazioni su un array C++ è vera?
- a) La sua dimensione può essere cambiata dinamicamente dopo la dichiarazione.
 - b) Memorizza elementi in locazioni di memoria contigue.
 - c) Può contenere elementi di tipi di dato diversi.
 - d) La funzione `size()` restituisce la sua dimensione.
20. Qual è il numero di confronti necessari nel caso peggiore per trovare un elemento in una lista concatenata semplice di n elementi?
- a) $\log n$
 - b) $n/2$
 - c) 1
 - d) n
-

Test 4

1. Cosa fa la seguente funzione `inverti` (supponendo che la riga mancante sia `*testa = prev;`)?

```
void inverti(Nodo** testa) {  
    Nodo* prev = nullptr;  
    Nodo* current = *testa;  
    Nodo* next = nullptr;  
    while (current != nullptr) {  
        next = current->succ;  
        current->succ = prev;  
        prev = current;  
        current = next;  
    }  
    // riga mancante
```

```
}
```

- a) Cancella tutti i nodi della lista.
- b) Inverte l'ordine dei nodi nella lista concatenata.
- c) Stampa la lista al contrario.
- d) Causa un errore di segmentazione.

2. Qual è l'output del codice seguente?

```
int i = 5, j = 10;  
int *p1 = &i;  
int *p2 = &j;  
*p1 = *p2;  
*p2 = 0;  
cout << i << " " << j;
```

- a) 5 10
- b) 10 0
- c) 5 0
- d) 10 10

3. Quale operazione è tipicamente associata a una Coda (Queue)?

- a) push
- b) top
- c) pop
- d) enqueue

4. Qual è la complessità della funzione seguente?

```
void func(int n) {  
    int count = 0;  
    for (int i = n / 2; i <= n; i++)  
        for (int j = 1; j <= n; j = 2 * j)  
            for (int k = 1; k <= n; k = k * 2)  
                count++;  
}
```

- a) $O(n^2)$
- b) $O(n \log_2 n)$
- c) $O(n \log n)$
- d) $O(n^3)$

5. Una classe che contiene almeno una funzione "pure virtual" è chiamata:

- a) Classe template
- b) Classe friend
- c) Classe astratta
- d) Classe virtuale

6. Qual è il modo corretto per aprire un file `log.txt` per aggiungere testo alla fine del file?

- a) `ofstream file("log.txt");`
- b) `ofstream file("log.txt", ios::in);`
- c) `ofstream file("log.txt", ios::app);`
- d) `ifstream file("log.txt", ios::app);`

7. In un Albero Binario di Ricerca (BST), per ogni nodo N:
 - a) Tutti i valori nel sottoalbero sinistro di N sono maggiori del valore di N.
 - b) Tutti i valori nel sottoalbero destro di N sono minori del valore di N.
 - c) Tutti i valori nel sottoalbero sinistro di N sono minori o uguali al valore di N.
 - d) I valori nei sottoalberi non hanno una relazione ordinata con N.
8. Cosa fa l'operatore `::` (Scope Resolution Operator) in C++?
 - a) Accede a membri di un oggetto tramite un puntatore.
 - b) Specifica a quale namespace o classe appartiene un membro.
 - c) Esegue una divisione tra interi.
 - d) Dichiara un riferimento.
9. L'ereditarietà `protected` fa sì che i membri `public` della classe base diventino...
 - a) `public` nella classe derivata.
 - b) `private` nella classe derivata.
 - c) `protected` nella classe derivata.
 - d) inaccessibili.
10. Quale delle seguenti affermazioni è vera per le stringhe C-style (`char*`)?
 - a) Hanno una dimensione dinamica gestita automaticamente.
 - b) Terminano con il carattere nullo `\0`.
 - c) Supportano l'operatore `+` per la concatenazione.
 - d) Sono più sicure delle `std::string`.
11. Quale di queste operazioni è più lenta in una Pila (Stack) implementata con una lista concatenata?
 - a) `push` (inserimento)
 - b) `pop` (rimozione)
 - c) `top` (lettura cima)
 - d) Tutte le operazioni hanno complessità $O(1)$.
12. Cosa succede se si usa `delete` su un puntatore che non è stato allocato con `new`?
 - a) Nulla, l'operazione è sicura.
 - b) Viene generato un errore a tempo di compilazione.
 - c) Il comportamento è indefinito e può portare a crash del programma.
 - d) Viene liberata memoria a caso.
13. Qual è la complessità temporale della traversata di un albero binario con n nodi (es. in-ordine, pre-ordine, post-ordine)?
 - a) $O(1)$
 - b) $O(\log n)$
 - c) $O(n)$
 - d) $O(n \log n)$
14. Un costruttore di una classe:
 - a) Deve avere un tipo di ritorno `void`.
 - b) Non può avere parametri.
 - c) Ha lo stesso nome della classe.
 - d) È sempre `private`.

15. Per inserire un nodo dopo un nodo `p` esistente in una lista concatenata semplice, quale operazione è corretta (`nuovoNodo` è il nodo da inserire)?

- a) `nuovoNodo->succ = p->succ; p->succ = nuovoNodo;`
- b) `p->succ = nuovoNodo; nuovoNodo->succ = p;`
- c) `nuovoNodo->succ = p; p = nuovoNodo;`
- d) `p->succ = nuovoNodo;`

16. Quale delle seguenti è una buona ragione per passare un oggetto a una funzione per riferimento (&) invece che per valore?

- a) Per garantire che la funzione non possa modificare l'oggetto originale.
- b) Per evitare il costo di copiare un oggetto di grandi dimensioni.
- c) Perché è l'unico modo per passare oggetti.
- d) Per creare una copia locale dell'oggetto.

17. Cosa fa `getline(cin, str);`?

- a) Legge un singolo carattere da `cin` e lo mette in `str`.
- b) Legge una riga intera da `cin`, inclusi gli spazi, e la mette nella stringa `str`.
- c) Legge una parola da `cin` (fino al primo spazio) e la mette in `str`.
- d) Non compila, la sintassi è errata.

18. Un `template <class T>` è un esempio di:

- a) Polimorfismo a tempo di esecuzione.
- b) Polimorfismo a tempo di compilazione (o parametrico).
- c) Incapsulamento.
- d) Ereditarietà.

19. Qual è l'altezza di un albero binario completo con n nodi?

- a) $O(n)$
- b) $O(\log n)$
- c) $O(1)$
- d) $O(n^2)$

20. Qual è la complessità del seguente codice?

```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```

- a) $O(N)$
- b) $O(N^2)$
- c) $O(\sqrt{N})$
- d) $O(\log N)$