


Giảng viên ra đề: (Chữ ký và Họ tên)	(Ngày ra đề)	Người phê duyệt: (Chữ ký và họ tên)	(Ngày duyệt đề)
--	--------------	---	-----------------

<div></div> <div>TRƯỜNG ĐH BÁCH KHOA - ĐHQG-HCM</div> <div><u>KHOA KH & KT MÁY TÍNH</u></div>	THI CUỐI KỲ		Học kỳ / Năm học		2	2023-2024
			Ngày thi		22-05-2024	
	Môn học	Kỹ thuật lập trình				
	Mã môn học	CO1027				
	Thời lượng	90 phút	Mã đề	2350		
Ghi chú: <ul style="list-style-type: none">- Đề thi có tổng cộng 60 câu.- Sinh viên được phép sử dụng một tờ A4 tài liệu tham khảo viết tay và máy tính cầm tay.- Các thư viện cơ bản đã được include đầy đủ.- Nộp lại đề sau khi kiểm tra/thi.						

Câu 1. [L.O.2.3] Dòng nào trong đoạn code dưới đây là KHÔNG hợp lệ:

```
int a = 10;
int b = 11;
const int* p = &a; // Line 3
*p = 100; // Line 4
p = &b; // Line 5
```

- (A) Dòng 3 (B) Dòng 5 (C) Dòng 4 (D) Tất cả đều hợp lệ

Cho chương trình như sau; giả sử rằng các thư viện cần thiết đã được **include**. Dùng đoạn mã này cho các câu hỏi từ 2 - 3.

```
/*Mark-1*/ A {
    A(int value=9): value_(value){ cout << "A, "; }
    ~A(){ cout << "~A, "; }
/*Mark-2*/
    int value_;
};
int main() {
    cout << "[";
    {A a; {A a; cout << "x, ";} cout << a.value_ << ", "; }
    cout << "]" ; return 0;
}
```

Câu 2. Thay thế /*Mark-1*/ bởi **struct** và /*Mark-2*/ bởi **public**:. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- (A) Có lỗi trong **main**, vì có hai biến cùng tên là **a**. (B) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private.
 (C) Kết quả in ra là: [A, A, x] (D) Tất cả các lựa chọn khác đều sai.

Câu 3. Thay thế /*Mark-1*/ bởi **class** và /*Mark-2*/ bởi **public**:. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- (A) Có lỗi trong **main**, vì có hai biến cùng tên là **a**. (B) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private.
 (C) Kết quả in ra là: [A, A, x, ~A, 9, ~A,] (D) Tất cả các lựa chọn khác đều sai.

Thông tin sau áp dụng cho câu hỏi 4:

```
class Triangle {
public:
    static int numOfTriangle;
private:
    int a;
    int b;
    int c;
public:
    Triangle(int a, int b, int c) : a(a), b(b), c(c) {
        numOfTriangle++;
    }
    ~Triangle() {
        numOfTriangle--;
        cout << 3;
    }
};

int Triangle::numOfTriangle = 0;
int main() {
    Triangle aList[5];
    cout << (*aList).numOfTriangle;
    return 0;
}
```

Câu 4. [L.O.3.1] Cho biết kết quả in ra màn hình của chương trình:

Ⓐ 5

Ⓑ 53

Ⓒ 533333

Ⓓ Lỗi biên dịch

Câu 5. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int array1[] = {1, 2, 3, 4, 5};
int array2[] = {6, 7, 8, 9, 10};
int temp, result = 0;
int main() {
    for (temp = 0; temp < 5; temp++) {
        result += array1[temp];
    }
    for (; temp < 4; temp++) {
        result += array2[temp];
    }
    cout << result;
    return 0;
}
```

Ⓐ 55

Ⓑ 15

Ⓒ 45

Ⓓ 35

Câu 6. [L.O.3.1] Cho đoạn code dưới đây:

```
class Node {  
public:  
    Node() { cout << "Normal Constructor called\n"; }  
    Node(const Node &t) { cout << "Copy constructor called\n"; }  
};  
  
int main() {  
    Node *t1, *t2; // Line 1  
    t1 = new Node(); // Line 2  
    t2 = new Node(*t1); // Line 3  
    Node t3 = *t1; // Line 4  
    return 0;  
}
```

Chọn câu SAI:

- (A) Normal constructor được gọi ở dòng 2 (B) Normal constructor được gọi ở dòng 1
(C) Copy constructor được gọi ở dòng 3 (D) Copy constructor được gọi ở dòng 4

Câu 7. [L.O.3.1] Khi xây dựng một lớp trong C++, nếu không xác định cụ thể, các thuộc tính và phương thức của lớp đó sẽ có tầm vực mặc định là:

- (A) protected (B) private (C) public (D) package

Câu 8. [L.O.3.2] Cho lớp A là cha của lớp B, và biến a có kiểu A, biến b có kiểu B.

Phát biểu gán 1: a = b;

Phát biểu gán 2: b = a;

- (A) Phát biểu 1 đúng, 2 sai (B) Phát biểu 1 sai, 2 đúng
(C) Cả hai phát biểu gán đều đúng (D) Cả hai phát biểu gán đều đúng

Câu 9. Biểu tượng hình thoi trong sơ đồ luồng (flowchart) biểu diễn cho:

- (A) Một điểm nơi mà có một câu hỏi được đặt và luồng được phân nhánh dựa vào câu trả lời (B) Một liên kết để kết nối các phần khác nhau của sơ đồ luồng
(C) Một quy trình hoặc một hành động cụ thể (D) Điểm bắt đầu hoặc điểm kết thúc của luồng đó

Câu 10. [L.O.2.2] Trong C++, cho các string s1, s2 và s3 được khai báo như sau:

```
string s1 = "Hello_World";  
string s2(s1, 3);  
string s3(s2, 1, 4);
```

Giá trị của string s3 là?

- (A) o_Wo (B) orld (C) ello (D) _Wor

Câu 11. [L.O.3.1] Câu nào sau đây mô tả chính xác hàm ảo trong lập trình hướng đối tượng?

- (A) Một hàm có thể được ghi đè bởi (các) lớp kế thừa để cung cấp cách triển khai cụ thể.
(B) Một hàm được khai báo bằng từ khóa virtual chỉ có thể được truy cập trong lớp nơi nó được khai báo.
(C) Hàm được gọi tự động khi một đối tượng được tạo.
(D) Một hàm có thể được gọi mà không cần một thể hiện đối tượng.

Câu 12. [L.O.3.2] Thế nào được gọi là hiện tượng nạp chồng (overriding)?

- (A) Hiện tượng lớp con kế thừa định nghĩa một hàm hoàn toàn giống lớp cha
- (B) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên nhưng khác kiểu với một hàm ở lớp cha
- (C) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng kiểu với một hàm ở lớp cha nhưng khác đối số
- (D) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng các đối số nhưng khác kiểu với một hàm ở lớp cha

Câu 13. [L.O.2.2] Trong C++, ta khai báo một mảng số nguyên hai chiều như sau: `int arr[5][7];` Kích thước của một giá trị số nguyên là 4 byte. Địa chỉ ô nhớ bắt đầu của `arr` là `0x70`, địa chỉ vùng nhớ của `arr[3][4]` là bao nhiêu?

- (A) `0xa0`
- (B) `0xd4`
- (C) `0xfc`
- (D) `0x8c`

Câu 14. [L.O.1.1] Chọn phát biểu đúng về mô tả giải thuật:

- (A) Cần phải dùng mã lập trình
- (B) Cần phải dùng lưu đồ (flowchart)
- (C) Cần phải dùng mã giả (pseudocode)
- (D) Có nhiều cách khác nhau để mô tả giải thuật

Câu 15. [L.O.2.3] Cách truy xuất vào một thành phần của đối tượng con trở this?

- (A) `this_x`
- (B) `(*this)->x`
- (C) `this->x`
- (D) `this.x`

Câu 16. [L.O.3.1] Một đối tượng trong lập trình hướng đối tượng bao gồm:

- (A) Hằng số và biến
- (B) Cấu trúc dữ liệu
- (C) Toán tử và hàm
- (D) Thuộc tính và phương thức

Câu 17. [L.O.2.1] Cho hiện thực hóa 1 hàm như sau:

```
void func(int* const list, int arraySize) {  
    for (int i=0; i<arraySize; i++)  
        list[i] = list[i] + 1;  
}  
  
int main() {  
    int numbers[5] = {5, 2, 7, 7, 1};  
    func(numbers, 5);  
    for (int i=0; i < 5; i++)  
        cout << numbers[i];  
    return 0;  
}
```

The value of array *numbers* when running the *main()* function is:

- (A) {5, 2, 7, 7, 1}
- (B) {6, 3, 8, 8, 2}
- (C) Lỗi khi chạy
- (D) Lỗi biên dịch

Câu 18. [L.O.3.1] Chọn phát biểu SAI trong các phát biểu sau:

- (A) Một class chỉ có thể có duy nhất một hàm hủy (destructor)
- (B) Một class có thể có nhiều hàm khởi tạo (constructor)
- (C) Hàm khởi tạo có tham số truyền vào, nhưng hàm hủy thì không
- (D) Hàm khởi tạo có thể là một hàm ảo (virtual function) trong abstract class

Câu 19. [L.O.3.2] Trong lập trình hướng đối tượng (OOP), encapsulation có ý nghĩa gì?

- (A) Quy tắc cho phép một đối tượng che giấu thông tin bên trong và chỉ cho phép truy cập thông qua các phương thức
- (B) Quy tắc cho phép một đối tượng kế thừa các thuộc tính và phương thức từ một đối tượng khác
- (C) Quy tắc cho phép một đối tượng kết nối với các đối tượng khác
- (D) Quy tắc cho phép một đối tượng tự quản lý bộ nhớ

Câu 20. [L.O.2.3] Cách khai báo biến tham chiếu ref (reference variable) trong C++ là gì?

- (A) ref *int; (B) ref int; (C) int *ref; (D) int &ref;

Câu 21. [L.O.2.3] Hãy cho biết kết quả xuất ra màn hình của đoạn chương trình sau:

```
int x=1;
int *p=&x;
int &q=x;
cout << (q == *p);
```

- (A) 0 (B) 1 (C) Địa chỉ của x và giá trị của x
(D) Lỗi biên dịch

Câu 22. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main() {
    int a = 1, b = 10, c = 3;
    int *arr[3] = {&a, &b, &c};
    cout << *arr[*arr[0] * 2] - c;
    return 0;
}
```

- (A) 3 (B) 0 (C) Giá trị rác (D) Lỗi biên dịch

Câu 23. [L.O.2.4] Kết quả khi chạy đoạn chương trình sau là:

```
int A[] = {2, 4, 1, 6, 8, 3, 1, 10};
int *p = A;
while (p < A + 6) {
    cout << --*p++ << *p++;
    p++;
}
```

- (A) 1458 (B) 4168 (C) 3158 (D) 2367

Câu 24. [L.O.2.3] Trong đoạn code C++ dưới đây, giá trị nào sẽ được in ra sau khi chạy chương trình?

```
int main() {
    char str[] = "hcmut-cse";
    char *ptr = str;
    ptr += 5;
    cout << ptr;
    return 0;
}
```

- (A) t-cse (B) -cse
(C) cse (D) Chương trình bị lỗi biên dịch (Compilation Error)

Câu 25. [L.O.2.2] Kết quả in ra của đoạn chương trình sau, nếu chuỗi nhập vào ở lệnh cin là "I love BachKhoa":

```
string st;
cin >> st;
cout << st;
```

- (A) I love BachKhoa (B) I love (C) BachKhoa (D) I

Câu 26. [L.O.2.1] Kết quả của fun (2) là gì?

```
int fun(int n) {  
    if (n == 4)  
        return n;  
    else return 2*fun(n+1);  
}
```

- (A) 4 (B) 16 (C) 32 (D) 8

Câu 27. [L.O.2.3] Câu nào mô tả đúng nhất về memory leak?

- (A) Chương trình cố tình phân bổ nhiều bộ nhớ hơn mức cần thiết.
(B) Chương trình sử dụng bộ nhớ không hiệu quả, dẫn đến hiệu suất chậm hơn.
(C) Lỗi lập trình dẫn đến hỏng dữ liệu được lưu trữ trong bộ nhớ.
(D) Chương trình không giải phóng được bộ nhớ không sử dụng, khiến bộ nhớ khả dụng dần cạn kiệt.

Câu 28. [L.O.3.2] Class B sử dụng được các biến thành viên nào của class A:

```
class A {  
    int a, b;  
    protected:  
        double c, d;  
};  
class B : public A {  
};
```

- (A) a, b (B) c, d
(C) a, b, c, d (D) Không truy cập được biến nào

Câu 29. [L.O.2.2] Trong ngôn ngữ lập trình C++, để khai báo một mảng có kích thước cố định và lưu trữ các số nguyên, ta sử dụng cú pháp nào?

- (A) `array<int> myArray(10);` (B) `int myArray[10];`
(C) `array<int, 10> myArray;` (D) `int[] myArray = new int[10];`

Câu 30. [L.O.2.3] Kết quả của đoạn code C bên dưới là gì

```
int main() {  
    int arr[] = {1, 2, 3, 4, 5};  
    int *ptr = arr;  
    printf("%d", *(ptr + 3));  
    return 0;  
}
```

- (A) 2 (B) 5 (C) 4 (D) Lỗi runtime

Câu 31. [L.O.2.3] Trong lập trình C, con trỏ **void*** có thể đại diện cho cái gì?

- (A) Con trỏ đến địa chỉ của một biến kiểu void (B) Con trỏ đến một hàm có kiểu trả về là void
(C) Con trỏ đến một biến hằng (D) Con trỏ đến địa chỉ của kiểu dữ liệu bất kỳ

Câu 32. [L.O.2.1] Cho đoạn mã chương trình sau của một hàm đệ qui và cách sử dụng hàm:

```
void recur_func(int n) {  
    if (n<=1)  
        return;  
    cout << n << " -> ";  
    if(n % 2)  
        recur_func(3*n+1);  
    else  
        recur_func(n/2);  
}  
int main() {  
    int n = 5;  
    recur_func(n);  
    return 0;  
}
```

Kết quả khi chạy hàm *main()* là:

- Ⓐ 5 -> 16 -> 8 -> 4 -> 2 -> Ⓑ 5 -> 2 -> 16 -> 2 -> 4 ->
Ⓒ 5 -> 4 -> 2 -> 16 -> 8 -> Ⓓ Runtime error (Lỗi chạy chương trình)

Câu 33. [L.O.1.1] Hãy cho biết output của mã giả sau:

```
Set numList to [4, 6, 0, 1, 8 ,3, 9]  
Set res to numList[0]  
  
For i from 1 to length of numList - 1:  
    If numList[i] > res:  
        Set res to numList[i-1]  
End For  
  
Output res
```

- Ⓐ 8 Ⓑ 9
Ⓒ 3 Ⓓ Các đáp án khác đều sai

Câu 34. [L.O.1.1] Hãy cho biết output của mã giả sau:

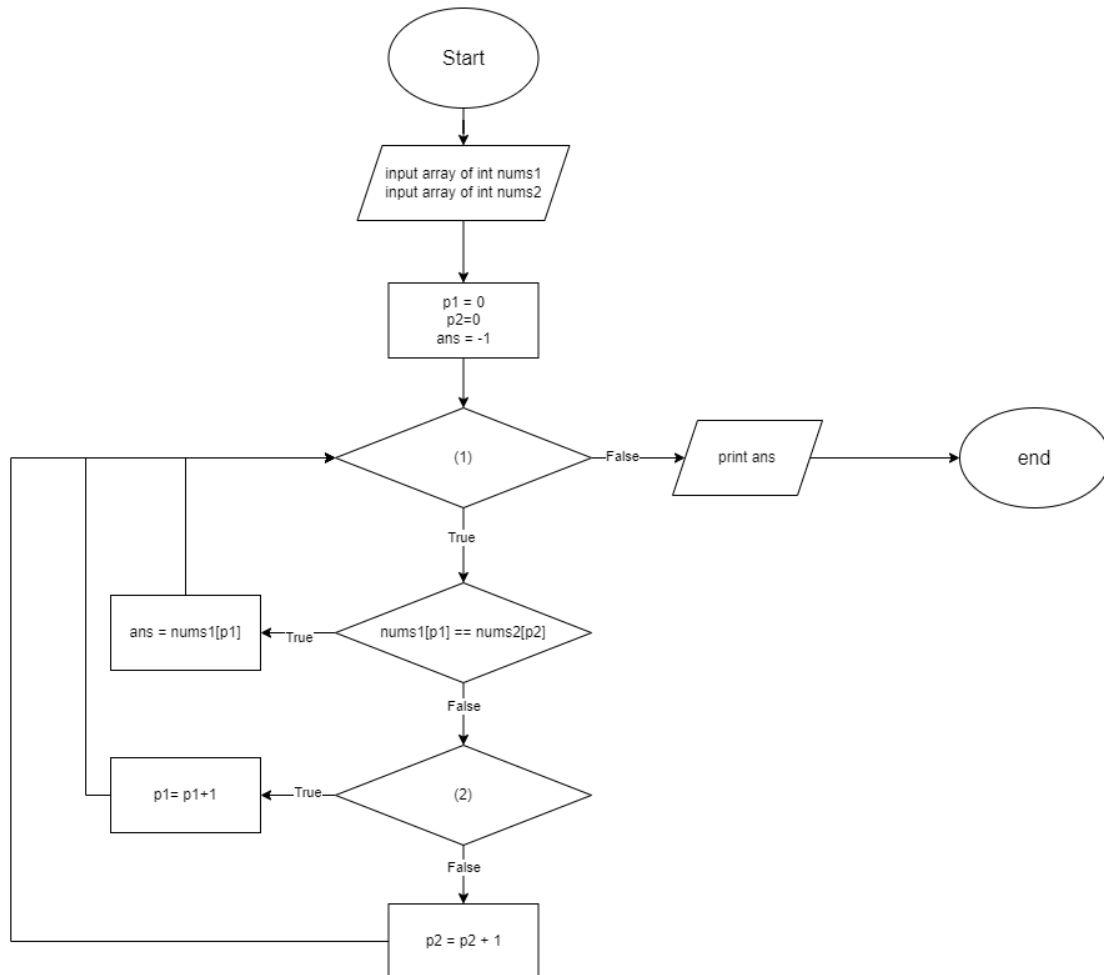
```
Set matrix to [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
Set finalMatrix to empty list  
  
For i from 0 to 2:  
    Set newRow to empty list  
    For j from 0 to 2:  
        Append matrix[2-j][i] to newRow  
    End For  
    Append newRow to finalMatrix  
End For
```

- Ⓐ [[9, 6, 3], [8, 5, 2], [7, 4, 1]] Ⓑ [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Ⓒ [[3, 6, 9], [2, 5, 8], [1, 4, 7]] Ⓓ [[1, 4, 7], [2, 5, 8], [3, 6, 9]]

Câu 35. [L.O.3.1] Điều nào sau đây mô tả đúng nhất các lớp trừu tượng?

- (A) Nếu một lớp có nhiều hơn một hàm ảo thì đó là lớp trừu tượng
- (B) Nếu một lớp có ít nhất một hàm ảo thuần túy thì đó là lớp trừu tượng
- (C) Nếu một lớp chỉ có một hàm ảo thuần túy thì đó là lớp trừu tượng
- (D) Nếu một lớp chỉ có tất cả các hàm ảo thuần túy thì đó là lớp trừu tượng

Câu 36. [L.O.1.2] Để thực hiện thuật toán tìm phần tử nhỏ nhất **cùng xuất hiện** trong cả 2 mảng được sắp xếp theo thứ tự **tăng dần**, một thuật toán được thiết kế theo flowchart sau, hãy điền vào vị trí (1) và (2) sao cho chương trình sẽ in ra giá trị phần tử thỏa đề nếu có và ngược lại in -1 (Lưu ý, nums1.size() là lấy số phần tử có trong mảng nums1)



- (A) (1): ans == -1 && p1 < nums1.size() && p2 < nums2.size() (2): nums1[p1] < nums2[p2]
- (B) (1): ans == -1 || (p1 < nums1.size() && p2 < nums2.size()) (2): nums1[p1] < nums2[p2]
- (C) (1): ans == -1 && p1 < nums1.size() && p2 < nums2.size() (2): nums1[p1] > nums2[p2]
- (D) (1): ans == -1 || (p1 < nums1.size() && p2 < nums2.size()) (2): nums1[p1] > nums2[p2]

Câu 37. [L.O.2.4] Kết quả chương trình sau là gì?

```

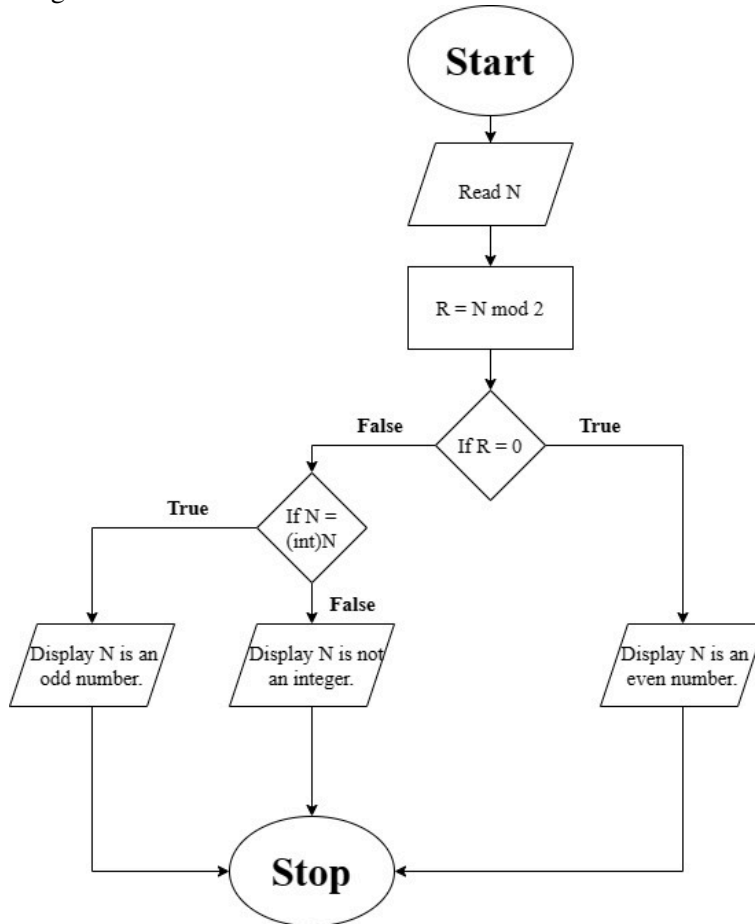
#define foo(a, b) ((a) > (b) ? (a) : (b))
int main() {
    int x = 10, y = 20;
    cout << foo(x, y) << endl;
    return 0;
}
  
```

- (A) Lỗi biên dịch
- (B) 10
- (C) 20
- (D) Lỗi runtime

Câu 38. [L.O.2.3] Mục đích của con trỏ trong lập trình là gì

- (A) Lưu trữ địa chỉ của biến
- (B) Lưu trữ giá trị của biến
- (C) Thực hiện các phép tính số học
- (D) Cấp phát bộ nhớ động

Câu 39. [L.O.1.2] Trường hợp nào không thể xảy ra với lưu đồ được cho bên dưới. Giả sử lưu đồ sẽ được hiện thực bằng C++.



- (A) Đầu vào: $N = 1$, Đầu ra: N is an odd number
- (B) Đầu vào: $N = 4$, Đầu ra: N is an even number
- (C) Đầu vào: $N = 3.5$, Đầu ra: N is not an integer
- (D) Tất cả các câu trên đều có thể xảy ra.

Câu 40. [L.O.1.2] Ký hiệu nào dưới đây trong lưu đồ (flowchart) được dùng để biểu thị output được trả về?

- (A) Hình bình hành
- (B) Hình chữ nhật
- (C) Hình tròn/ellipse
- (D) Hình thoi

Câu 41. [L.O.2.1] Độ phức tạp của hàm đệ quy sau là:

```
int recursiveFunction(int n) {  
    if (n <= 1) {  
        return 1;  
    }  
    return recursiveFunction(n - 1) + recursiveFunction(n - 2);  
}
```

- (A) $O(n!)$
- (B) $O(n^2)$
- (C) $O(2^n)$
- (D) $O(\log n)$

Câu 42. [L.O.2.2] Phát biểu nào sau đây là ĐÚNG về Struct trong C++

- (A) Tầm vực truy xuất mặc định của các phần tử trong struct là private
- (B) Struct không hỗ trợ kế thừa
- (C) Có thể thay đổi tầm vực truy xuất của các thành phần trong struct
- (D) Struct không thể có phương thức

Câu 43. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main(int argc, char const *argv[]) {  
    string str;  
    cin>>str;  
    cout<<str;  
    return 0;  
}
```

- (A) str
- (B) Chuỗi nhập vào bởi người dùng
- (C) Error (Lỗi)
- (D) Garbage value (giá trị rác)

Thông tin sau áp dụng cho câu hỏi 44:

```
int i = 0;  
  
class A {  
public:  
    A() {  
        i = 3;  
    }  
    ~A() {  
        i=10;  
    }  
};  
  
int foo() {  
    int &p = i;  
    A ob;  
    return p;  
}
```

Câu 44. [L.O.3.2] LỜI gọi hàm *foo()* sẽ trả về giá trị nào?

- (A) 0
- (B) 3
- (C) 10
- (D) Phương án khác

Các thông tin sau sử dụng cho câu 45 đến 46:

Một bản đồ được biểu diễn bằng mảng 2 chiều gồm *n_rows* (một biến kiểu int) là số hàng và *n_cols* (một biến kiểu int) là số cột. Mỗi phần tử của mảng là một đối tượng thuộc class MapElement. Trên bản đồ có 2 loại phần tử là đường đi (biểu diễn bằng class Path) và tường (biểu diễn bằng class Wall). Path có thể cho phép một nhân vật di chuyển được trong khi tường thì không cho phép nhân vật di chuyển.

```
/* a */  
MapElement *** map = new MapElement**[n_rows];  
for (int r = 0; r < n_rows; ++r) {  
    map[r] = new MapElement*[n_cols];  
    for (int c = 0; c < n_cols; ++c) {  
        map[r][c] = new Path();  
    }  
}
```

```

    }
}

/* b */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    map[r] = new MapElement*[n_cols];
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}

/* c */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Path();
    }
}

/* d */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}

```

Câu 45. [L.O.2.3] Biến map được sử dụng để biểu diễn bản đồ. Giả sử mảng 2 chiều cho bản đồ được cấp phát động. Cho yêu cầu cần khởi tạo một bản đồ mà tất cả các phần tử đều là Path. Biết rằng Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu này:

- | | |
|--|--|
| (A) Đoạn code bên dưới comment /* a */ | (B) Đoạn code bên dưới comment /* b */ |
| (C) Đoạn code bên dưới comment /* c */ | (D) Đoạn code bên dưới comment /* d */ |

Câu 46. [L.O.2.3] Biến map được sử dụng để biểu diễn bản đồ. Biết rằng: hiện tại bản đồ gồm tất cả các phần tử đều là Path; Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Cho yêu cầu: tạo ra một bản đồ gồm tất cả phần tử ở hàng đầu tiên đều là Wall. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu trên và không tạo ra rò rỉ bộ nhớ (memory leak):

```
/* a */
for (int i = 0; i < n_cols; ++i) {
    map[0][i] = new Wall();
}

/* b */
for (int i = 0; i < n_cols; ++i) {
    delete map[0][i];
    map[0][i] = new Wall();
}

/* c */
for (int i = 0; i < n_rows; ++i) {
    delete map[i][0];
    map[i][0] = new Wall();
}

/* d */
for (int i = 0; i < n_cols; ++i) {
    delete (*map[n_rows-1][i]);
    map[n_rows-1][i] = new Wall();
}
```

- (A) Đoạn code bên dưới comment /* b */
 (C) Đoạn code bên dưới comment /* c */

- (B) Đoạn code bên dưới comment /* a */
 (D) Đoạn code bên dưới comment /* d */

Các thông tin sau sử dụng cho câu 47. Một nhân vật có một túi đồ để chứa các vật phẩm có thể được thu thập trong khi di chuyển trên bản đồ. Class Character biểu diễn một nhân vật. Một Character có 2 thuộc tính private là hp và exp. Class Character cũng có các cặp hàm get, set để lấy và gán giá trị của thuộc tính tương ứng. Class BaseBag biểu diễn một túi đồ có mã nguồn được cho bên dưới. Abstract class BaseItem biểu diễn một vật phẩm. Túi đồ được hiện thực bằng danh sách liên kết, mỗi phần tử của danh sách là một Node, head và tail lần lượt là Node đầu tiên và Node cuối cùng của danh sách.

```
class Character {
private:
    int hp, exp;
public:
    int getHp() const;
    void setHp(int hp);
    int getExp() const;
    void setExp(int exp);
}

class BaseItem {
// Definition of class BaseItem
};

class Node {
```

```

friend class BaseBag;
private:
    BaseItem * item;
    Node * next;
public:
    Node(BaseItem * item, Node * next = NULL) {
        this->item = item;
        this->next = next;
    }
    ~Node() {}
};

```

```

class BaseBag {
private:
    Node * head, * tail;
    int capacity;
    int count;
protected:
    virtual void removeFirst() {
        if (count == 1) {
            delete head;
            head = NULL;
            tail = NULL;
        }
        else {
            /* r1 */
            /* r2 */
            /* r3 */
        }
        --count;
    }
public:
    BaseBag(int capacity) {
        this->capacity = capacity;
        this->count = 0;
        this->head = this->tail = NULL;
    }
};

```

Câu 47. [L.O.2.3] Phương thức removeFirst(Character * ch) của BaseBag thực hiện xóa vật phẩm đầu tiên của túi đồ nếu túi đồ có ít nhất 1 vật phẩm (giả sử khi được gọi, phương thức luôn có ít nhất 1 vật phẩm). Chọn đoạn mã hợp lệ để điền vào các ghi chú (comment) /* r1 */, /* r2 */, /* r3 */:

- (A) /* r1 */ Node * temp = **this**->head; /* r2 */ head = head->next; /* r3 */ delete temp;
- (B) /* r1 */ delete head; /* r2 */ head = head->next; /* r3 */ (không cần điền mã)
- (C) /* r1 */ head = head->next; /* r2 */ delete head; /* r3 */ (không cần điền mã)
- (D) /* r1 */ Node * temp = head; /* r2 */ delete head; /* r3 */ head = temp->next;

Các thông tin sau sử dụng cho câu 48 đến 50:

Cho struct PasswordInfo mô tả thông tin về một mật khẩu trong mảng arr_pwds[] cụ thể như sau:

```

struct PasswordInfo {
    char pwd[50];
    int freq;
    int first_index;
};

```

Trong đó:

- char pwd[50]: Chuỗi password
- int freq: tần suất xuất hiện của password trong mảng arr_pwds[]
- first_index: Vị trí index xuất hiện đầu tiên của password

Hàm void buildArrayUniquePassword>PasswordInfo arr_unique_pwds[], int & num_unique_pwds, const char * arr_pwds[], int num_pwds) để tạo ra mảng các PasswordInfo không trùng nhau từ mảng các password trong arr_pwds.

Tham số đầu vào gồm:

- PasswordInfo arr_unique_pwds[]: Mảng các PasswordInfo không trùng nhau
- int & num_unique_pwds: Giá trị tham chiếu số lượng password không trùng nhau
- arr_pwds và num_pwds lần lượt là mảng các password và số lượng phần tử của mảng

Nội dung hàm như sau:

```

void buildArrayUniquePassword>PasswordInfo arr_unique_pwds[], int &
num_unique_pwds, const char * arr_pwds[], int num_pwds) {
    bool first_meet;
    num_unique_pwds = 0;
    //First loop
    for (int i_f = 0; i_f < num_pwds; ++i_f) { // index_full
        first_meet = true;
        //Second loop
        for (int i_u = 0; i_u < num_unique_pwds; ++i_u) { // index_unique
            if (strcmp(arr_pwds[i_f], arr_unique_pwds[i_u].pwd) == 0) {
                ++arr_unique_pwds[i_u].freq;
                first_meet = false;
                break;
            }
        }
        //missing_stmt_a
        {
            strcpy(arr_unique_pwds[num_unique_pwds].pwd, arr_pwds[i_f]);
            arr_unique_pwds[num_unique_pwds].first_index = i_f;
            //missing_stmt_b
            ++num_unique_pwds;
        }
    }
}

```

Câu 48. [L.O.2.2] Nếu thiếu câu lệnh num_unique_pwds = 0 bên trong như hàm trên điều gì sẽ xảy ra?

- (A) Chương trình có thể sẽ gây ra lỗi
- (B) Chương trình sẽ chỉ lặp vòng lặp //Second loop qua đúng 1 lần tại phần tử đầu tiên của mảng arr_unique
- (C) Giá trị freq của PasswordInfo tương ứng sẽ luôn luôn không được cập nhật
- (D) Số lần lặp nhiều hơn nhưng kết quả vẫn chính xác

Câu 49. [L.O.2.2] Nội dung tại `//missing_stmt_a` có thể là:

- Ⓐ `if (first_meet)`
- Ⓑ `if(!first_meet)`
- Ⓒ `if (first_meet || num_unique_pwds)`
- Ⓓ `if (!first_meet || !num_unique_pwds)`

Câu 50. [L.O.2.2] Với nội dung đã chọn cho `//misng_stmt_a`, vậy nội dung câu lệnh có thể ở `//missing_stmt_b` là gì

- Ⓐ `arr_unique_pwds[num_unique_pwds].freq = 1;`
- Ⓑ `arr_unique_pwds[num_unique_pwds].freq ++;`
- Ⓒ `arr_unique_pwds[num_unique_pwds].freq = -1;`
- Ⓓ Đáp án A và B đều đúng

===== Kết thúc đề thi =====

Câu 1. C

Câu 2. D

Câu 3. B

Câu 4. D

Câu 5. B

Câu 6. B

Câu 7. B

Câu 8. A

Câu 9. A

Câu 10. A

Câu 11. A

Câu 12. A

Câu 13. B

Câu 14. D

Câu 15. C

Câu 16. D

Câu 17. B

Câu 18. D

Câu 19. A

Câu 20. D

Câu 21. B

Câu 22. B

Câu 23. A

Câu 24. B

Câu 25. D

Câu 26. B

Câu 27. D

Câu 28. B

Câu 29. B

Câu 30. C

Câu 31. D

Câu 32. A

Câu 33. C

Câu 34. A

Câu 35. B

Câu 36. A

Câu 37. C

Câu 38. A

Câu 39. D

Câu 40. A

Câu 41. C

Câu 42. C

Câu 43. B

Câu 44. B

Câu 45. A

Câu 46. A


Câu 47. A

Câu 48. A

Câu 49. A

Câu 50. A

Giảng viên ra đề: (Chữ ký và Họ tên)	(Ngày ra đề)	Người phê duyệt: (Chữ ký và họ tên)	(Ngày duyệt đề)
--	--------------	---	-----------------

<div></div> <div>TRƯỜNG ĐH BÁCH KHOA - ĐHQG-HCM</div> <div><u>KHOA KH & KT MÁY TÍNH</u></div>	THI CUỐI KỲ		Học kỳ / Năm học	2	2023-2024
			Ngày thi		22-05-2024
	Môn học	Kỹ thuật lập trình			
	Mã môn học	CO1027			
	Thời lượng	90 phút	Mã đề	2351	
Ghi chú: - Đề thi có tổng cộng 60 câu. - Sinh viên được phép sử dụng một tờ A4 tài liệu tham khảo viết tay và máy tính cầm tay. - Các thư viện cơ bản đã được include đầy đủ. - Nộp lại đề sau khi kiểm tra/thi.					

Câu 1. [L.O.2.2] Phát biểu nào sau đây là ĐÚNG về Struct trong C++

- ☐ (A) Struct không thể có phương thức
- ☐ (B) Tầm vực truy xuất mặc định của các phần tử trong struct là private
- ☐ (C) Struct không hỗ trợ kế thừa
- ☐ (D) Có thể thay đổi tầm vực truy xuất của các thành phần trong struct

Câu 2. [L.O.1.1] Hãy cho biết output của mã giả sau:

```
Set matrix to [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Set finalMatrix to empty list
```

```
For i from 0 to 2:
  Set newRow to empty list
  For j from 0 to 2:
    Append matrix[2-j][i] to newRow
  End For
  Append newRow to finalMatrix
End For
```

- ☐ (A) [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
- ☐ (B) [[9, 6, 3], [8, 5, 2], [7, 4, 1]]
- ☐ (C) [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
- ☐ (D) [[3, 6, 9], [2, 5, 8], [1, 4, 7]]

Câu 3. [L.O.2.3] Hãy cho biết kết quả xuất ra màn hình của đoạn chương trình sau:

```
int x=1;
int *p=&x;
int &q=x;
cout << (q == *p);
```

- ☐ (A) Lỗi biên dịch
- ☐ (B) 0
- ☐ (C) 1
- ☐ (D) Địa chỉ của x và giá trị của x

Câu 4. [L.O.2.3] Cách khai báo biến tham chiếu ref (reference variable) trong C++ là gì?

- ☐ (A) int &ref;
- ☐ (B) ref *int;
- ☐ (C) ref int;
- ☐ (D) int *ref;

Câu 5. [L.O.3.1] Chọn phát biểu SAI trong các phát biểu sau:

- Ⓐ Hàm khởi tạo có thể là một hàm ảo (virtual function) trong abstract class
- Ⓑ Một class chỉ có thể có duy nhất một hàm hủy (destructor)
- Ⓒ Một class có thể có nhiều hàm khởi tạo (constructor)
- Ⓓ Hàm khởi tạo có tham số truyền vào, nhưng hàm hủy thì không

Thông tin sau áp dụng cho câu hỏi 6:

```
int i = 0;
```

```
class A {  
public:  
    A() {  
        i = 3;  
    }  
    ~A() {  
        i=10;  
    }  
};
```

```
int foo() {  
    int &p = i;  
    A ob;  
    return p;  
}
```

Câu 6. [L.O.3.2] Lời gọi hàm *foo()* sẽ trả về giá trị nào?

- Ⓐ Phương án khác
- Ⓑ 0
- Ⓒ 3
- Ⓓ 10

Câu 7. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int array1[] = {1, 2, 3, 4, 5};  
int array2[] = {6, 7, 8, 9, 10};  
int temp, result = 0;  
int main() {  
    for (temp = 0; temp < 5; temp++) {  
        result += array1[temp];  
    }  
    for (; temp < 4; temp++) {  
        result += array2[temp];  
    }  
    cout << result;  
    return 0;  
}
```

- Ⓐ 55
- Ⓑ 45
- Ⓒ 15
- Ⓓ 35

Câu 8. [L.O.2.1] Cho đoạn mã chương trình sau của một hàm đệ qui và cách sử dụng hàm:

```
void recur_func(int n) {
    if (n<=1)
        return;
    cout << n << " -> ";
    if(n % 2)
        recur_func(3*n+1);
    else
        recur_func(n/2);
}
int main() {
    int n = 5;
    recur_func(n);
    return 0;
}
```

Kết quả khi chạy hàm *main()* là:

- (A) 5 -> 16 -> 8 -> 4 -> 2 -> (B) 5 -> 4 -> 2 -> 16 -> 8 ->
(C) 5 -> 2 -> 16 -> 2 -> 4 -> (D) Runtime error (Lỗi chạy chương trình)

Câu 9. [L.O.3.2] Thế nào được gọi là hiện tượng nạp chồng (overriding)?

- (A) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng các đối số nhưng khác kiểu với một hàm ở lớp cha
(B) Hiện tượng lớp con kế thừa định nghĩa một hàm hoàn toàn giống lớp cha
(C) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên nhưng khác kiểu với một hàm ở lớp cha
(D) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng kiểu với một hàm ở lớp cha nhưng khác đối số

Cho chương trình như sau; giả sử rằng các thư viện cần thiết đã được **include**. Dùng đoạn mã này cho các câu hỏi từ 10 - 11.

```
/*Mark-1*/ A {
    A(int value=9): value_(value){ cout << "A, "; }
    ~A(){ cout << "~A, "; }
/*Mark-2*/
    int value_;
};
int main() {
    cout << "[";
    {A a; {A a; cout << "x, ";} cout << a.value_ << ", "; }
    cout << "]" ; return 0;
}
```

Câu 10. Thay thế */*Mark-1*/* bởi **struct** và */*Mark-2*/* bởi **public**. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- (A) Có lỗi trong **main**, vì có hai biến cùng tên là **a**. (B) Kết quả in ra là: [A, A, x]
(C) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private. (D) Tất cả các lựa chọn khác đều sai.

Câu 11. Thay thế /*Mark-1*/ bởi **class** và /*Mark-2*/ bởi **public**:. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- (A) Có lỗi trong **main**, vì có hai biến cùng tên là **a**. (B) Kết quả in ra là: [A, A, x, ~A, 9, ~A,]
(C) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private. (D) Tất cả các lựa chọn khác đều sai.

Câu 12. [L.O.3.1] Một đối tượng trong lập trình hướng đối tượng bao gồm:

- (A) Thuộc tính và phương thức (B) Hằng số và biến
(C) Cấu trúc dữ liệu (D) Toán tử và hàm

Câu 13. [L.O.3.2] Class B sử dụng được các biến thành viên nào của class A:

```
class A {  
    int a, b;  
    protected:  
        double c, d;  
};  
class B : public A {  
};
```

- (A) Không truy cập được biến nào (B) a, b
(C) c, d (D) a, b, c, d

Thông tin sau áp dụng cho câu hỏi 14:

```
class Triangle {  
public:  
    static int numOfTriangle;  
private:  
    int a;  
    int b;  
    int c;  
public:  
    Triangle(int a, int b, int c) : a(a), b(b), c(c) {  
        numOfTriangle++;  
    }  
    ~Triangle() {  
        numOfTriangle--;  
        cout << 3;  
    }  
};  
  
int Triangle::numOfTriangle = 0;  
int main() {  
    Triangle aList[5];  
    cout << (*aList).numOfTriangle;  
    return 0;  
}
```

Câu 14. [L.O.3.1] Cho biết kết quả in ra màn hình của chương trình:

- (A) Lỗi biên dịch (B) 5 (C) 53 (D) 533333

Câu 15. [L.O.3.1] Câu nào sau đây mô tả chính xác hàm ảo trong lập trình hướng đối tượng?

- (A) Một hàm có thể được gọi mà không cần một thể hiện đối tượng.
- (B) Một hàm có thể được ghi đè bởi (các) lớp kế thừa để cung cấp cách triển khai cụ thể.
- (C) Một hàm được khai báo bằng từ khóa virtual chỉ có thể được truy cập trong lớp nơi nó được khai báo.
- (D) Hàm được gọi tự động khi một đối tượng được tạo.

Câu 16. [L.O.2.2] Trong ngôn ngữ lập trình C++, để khai báo một mảng có kích thước cố định và lưu trữ các số nguyên, ta sử dụng cú pháp nào?

- (A) `int[] myArray = new int[10];`
- (B) `array<int> myArray(10);`
- (C) `int myArray[10];`
- (D) `array<int, 10> myArray;`

Câu 17. Biểu tượng hình thoi trong sơ đồ luồng (flowchart) biểu diễn cho:

- (A) Điểm bắt đầu hoặc điểm kết thúc của luồng đó
- (B) Một điểm nơi mà có một câu hỏi được đặt và luồng được phân nhánh dựa vào câu trả lời
- (C) Một liên kết để kết nối các phần khác nhau của sơ đồ luồng
- (D) Một quy trình hoặc một hành động cụ thể

Câu 18. [L.O.2.3] Cách truy xuất vào một thành phần của đối tượng con trở this?

- (A) `this.x`
- (B) `this_x`
- (C) `(*this)->x`
- (D) `this->x`

Câu 19. [L.O.2.4] Kết quả khi chạy đoạn chương trình sau là:

```
int A[] = {2, 4, 1, 6, 8, 3, 1, 10};
int *p = A;
while (p < A + 6) {
    cout << --*p++ << *p++;
    p++;
}
```

- (A) 2367
- (B) 1458
- (C) 4168
- (D) 3158

Câu 20. [L.O.1.2] Ký hiệu nào dưới đây trong lưu đồ (flowchart) được dùng để biểu thị output được trả về?

- (A) Hình thoi
- (B) Hình bình hành
- (C) Hình chữ nhật
- (D) Hình tròn/ellipse

Câu 21. [L.O.2.1] Độ phức tạp của hàm đệ quy sau là:

```
int recursiveFunction(int n) {
    if (n <= 1) {
        return 1;
    }
    return recursiveFunction(n - 1) + recursiveFunction(n - 2);
}
```

- (A) $O(\log n)$
- (B) $O(n!)$
- (C) $O(n^2)$
- (D) $O(2^n)$

Câu 22. [L.O.2.4] Kết quả chương trình sau là gì?

```
#define foo(a, b) ((a) > (b) ? (a) : (b))
int main() {
    int x = 10, y = 20;
    cout << foo(x, y) << endl;
    return 0;
}
```

- (A) Lỗi runtime
- (B) Lỗi biên dịch
- (C) 10
- (D) 20

Câu 23. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main() {  
    int a = 1, b = 10, c = 3;  
    int *arr[3] = {&a, &b, &c};  
    cout << *arr[*arr[0] * 2] - c;  
    return 0;  
}
```

- (A) Lỗi biên dịch (B) 3 (C) 0 (D) Giá trị rác

Câu 24. [L.O.2.3] Mục đích của con trỏ trong lập trình là gì

- (A) Cấp phát bộ nhớ động (B) Lưu trữ địa chỉ của biến
(C) Lưu trữ giá trị của biến (D) Thực hiện các phép tính số học

Câu 25. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main(int argc, char const *argv[]) {  
    string str;  
    cin>>str;  
    cout<<str;  
    return 0;  
}
```

- (A) str (B) Error (Lỗi)
(C) Chuỗi nhập vào bởi người dùng (D) Garbage value (giá trị rác)

Câu 26. [L.O.2.3] Câu nào mô tả đúng nhất về memory leak?

- (A) Chương trình không giải phóng được bộ nhớ không sử dụng, khiến bộ nhớ khả dụng dần cạn kiệt.
(B) Chương trình cố tình phân bổ nhiều bộ nhớ hơn mức cần thiết.
(C) Chương trình sử dụng bộ nhớ không hiệu quả, dẫn đến hiệu suất chậm hơn.
(D) Lỗi lập trình dẫn đến hỏng dữ liệu được lưu trữ trong bộ nhớ.

Câu 27. [L.O.2.2] Kết quả in ra của đoạn chương trình sau, nếu chuỗi nhập vào ở lệnh cin là "I love BachKhoa":

```
string st;  
cin >> st;  
cout << st;
```

- (A) I (B) I love BachKhoa (C) I love (D) BachKhoa

Câu 28. [L.O.3.1] Điều nào sau đây mô tả đúng nhất các lớp trừu tượng?

- (A) Nếu một lớp chỉ có tất cả các hàm ảo thuần túy thì đó là lớp trừu tượng
(B) Nếu một lớp có nhiều hơn một hàm ảo thì đó là lớp trừu tượng
(C) Nếu một lớp có ít nhất một hàm ảo thuần túy thì đó là lớp trừu tượng
(D) Nếu một lớp chỉ có một hàm ảo thuần túy thì đó là lớp trừu tượng

Câu 29. [L.O.2.3] Trong lập trình C, con trỏ **void*** có thể đại diện cho cái gì?

- (A) Con trỏ đến địa chỉ của kiểu dữ liệu bất kỳ (B) Con trỏ đến địa chỉ của một biến kiểu void
(C) Con trỏ đến một hàm có kiểu trả về là void (D) Con trỏ đến một biến hằng

Câu 30. [L.O.1.1] Chọn phát biểu đúng về mô tả giải thuật:

- (A) Có nhiều cách khác nhau để mô tả giải thuật (B) Cần phải dùng mã lập trình
(C) Cần phải dùng lưu đồ (flowchart) (D) Cần phải dùng mã giả (pseudocode)

Câu 31. [L.O.3.2] Trong lập trình hướng đối tượng (OOP), encapsulation có ý nghĩa gì?

- (A) Quy tắc cho phép một đối tượng tự quản lý bộ nhớ
- (B) Quy tắc cho phép một đối tượng che dấu thông tin bên trong và chỉ cho phép truy cập thông qua các phương thức
- (C) Quy tắc cho phép một đối tượng kế thừa các thuộc tính và phương thức từ một đối tượng khác
- (D) Quy tắc cho phép một đối tượng kết nối với các đối tượng khác

Câu 32. [L.O.2.2] Trong C++, ta khai báo một mảng số nguyên hai chiều như sau: `int arr[5][7];`
Kích thước của một giá trị số nguyên là 4 byte. Địa chỉ ô nhớ bắt đầu của arr là 0×70 , địa chỉ vùng nhớ của `arr[3][4]` là bao nhiêu?

- (A) $0 \times 8c$
- (B) $0 \times a0$
- (C) $0 \times d4$
- (D) $0 \times fc$

Câu 33. [L.O.3.1] Cho đoạn code dưới đây:

```
class Node {
public:
    Node() { cout << "Normal Constructor called\n"; }
    Node(const Node &t) { cout << "Copy constructor called\n"; }
};

int main() {
    Node *t1, *t2; // Line 1
    t1 = new Node(); // Line 2
    t2 = new Node(*t1); // Line 3
    Node t3 = *t1; // Line 4
    return 0;
}
```

Chọn câu SAI:

- (A) Copy constructor được gọi ở dòng 4
- (B) Normal constructor được gọi ở dòng 2
- (C) Normal constructor được gọi ở dòng 1
- (D) Copy constructor được gọi ở dòng 3

Câu 34. [L.O.3.2] Cho lớp A là cha của lớp B, và biến a có kiểu A, biến b có kiểu B.

Phát biểu gán 1: `a = b;`

Phát biểu gán 2: `b = a;`

- (A) Cả hai phát biểu gán đều đúng
- (B) Phát biểu 1 đúng, 2 sai
- (C) Phát biểu 1 sai, 2 đúng
- (D) Cả hai phát biểu gán đều đúng

Câu 35. [L.O.2.3] Trong đoạn code C++ dưới đây, giá trị nào sẽ được in ra sau khi chạy chương trình?

```
int main() {
    char str[] = "hcmut-cse";
    char *ptr = str;
    ptr += 5;
    cout << ptr;
    return 0;
}
```

- (A) t-cse
- (B) cse
- (C) -cse
- (D) Chương trình bị lỗi biên dịch (Compilation Error)

Câu 36. [L.O.2.1] Cho hiện thực hóa 1 hàm như sau:

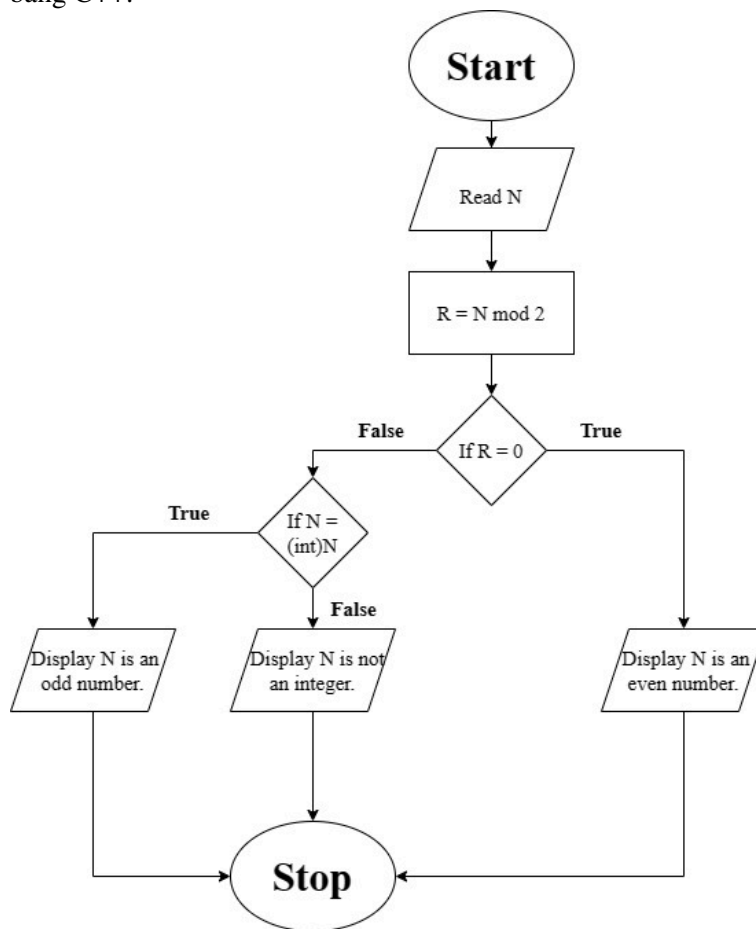
```
void func(int* const list, int arraySize) {
    for (int i=0; i<arraySize; i++)
        list[i] = list[i] + 1;
}

int main() {
    int numbers[5] = {5, 2, 7, 7, 1};
    func(numbers, 5);
    for (int i=0; i < 5; i++)
        cout << numbers[i];
    return 0;
}
```

The value of array *numbers* when running the *main()* function is:

- Ⓐ {5, 2, 7, 7, 1} Ⓑ Lỗi khi chạy Ⓒ {6, 3, 8, 8, 2} Ⓓ Lỗi biên dịch

Câu 37. [L.O.1.2] Trường hợp nào không thể xảy ra với lưu đồ được cho bên dưới. Giả sử lưu đồ sẽ được hiện thực bằng C++.



- Ⓐ Đầu vào: N = 1, Đầu ra: N is an odd number Ⓑ Đầu vào: N = 3.5, Đầu ra: N is not an integer
 Ⓒ Đầu vào: N = 4, Đầu ra: N is an even number Ⓓ Tất cả các câu trên đều có thể xảy ra.

Câu 38. [L.O.2.2] Trong C++, cho các string s1, s2 và s3 được khai báo như sau:

```
string s1 = "Hello_World";  
string s2(s1, 3);  
string s3(s2, 1, 4);
```

Giá trị của string s3 là?

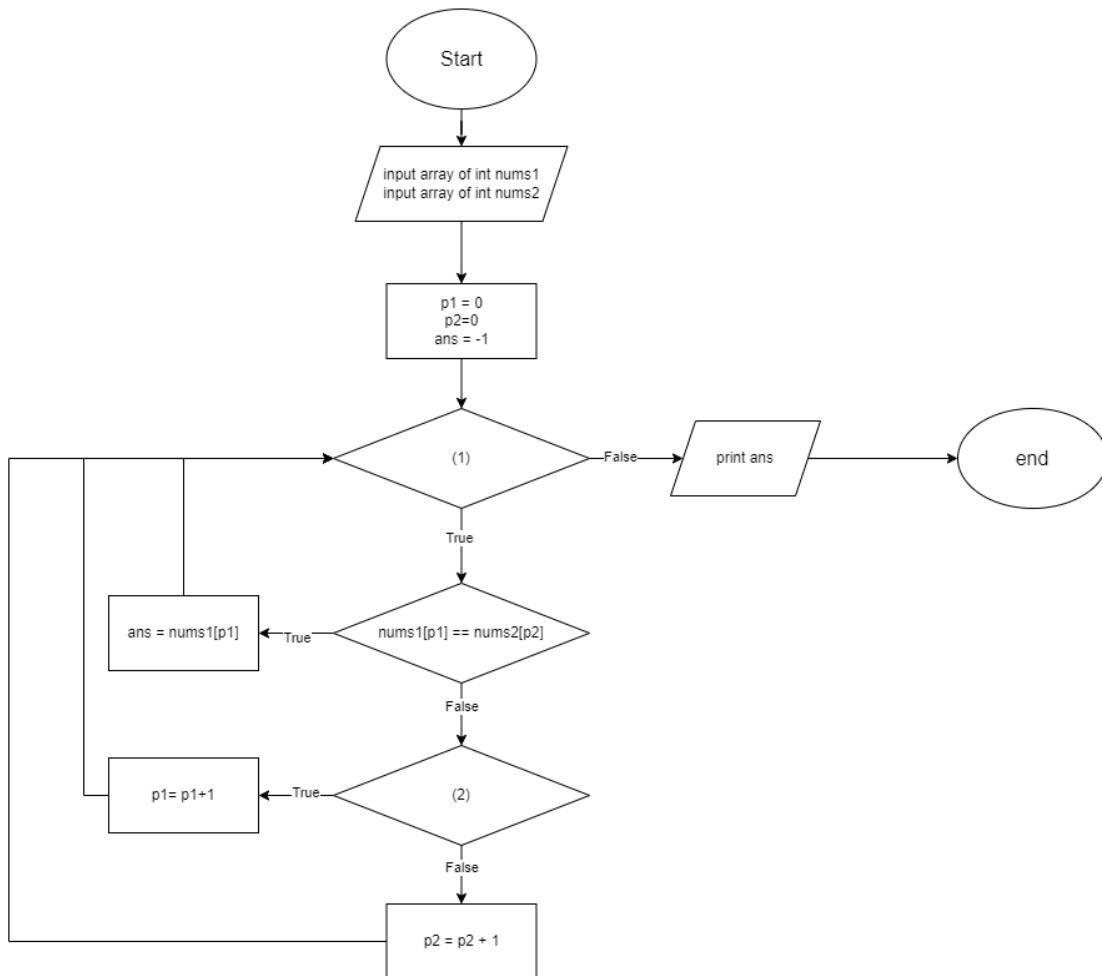
- ☐ (A) _Wor ☐ (B) o_Wo ☐ (C) orld ☐ (D) ello

Câu 39. [L.O.2.3] Kết quả của đoạn code C bên dưới là gì

```
int main() {  
    int arr[] = {1, 2, 3, 4, 5};  
    int *ptr = arr;  
    printf("%d", *(ptr + 3));  
    return 0;  
}
```

- ☐ (A) Lỗi runtime ☐ (B) 2 ☐ (C) 5 ☐ (D) 4

Câu 40. [L.O.1.2] Để thực hiện thuật toán tìm phần tử nhỏ nhất *cùng xuất hiện* trong cả 2 mảng được sắp xếp theo thứ tự *tăng dần*, một thuật toán được thiết kế theo flowchart sau, hãy điền vào vị trí (1) và (2) sao cho chương trình sẽ in ra giá trị phần tử thỏa đề nếu có và ngược lại in -1 (Lưu ý, `nums1.size()` là lấy số phần tử có trong mảng `nums1`)



- (A) (1): `ans == -1 || (p1 < nums1.size() && p2 < nums2.size())` (2): `nums1[p1] > nums2[p2]`
 (B) (1): `ans == -1 && p1 < nums1.size() && p2 < nums2.size()` (2): `nums1[p1] < nums2[p2]`
 (C) (1): `ans == -1 || (p1 < nums1.size() && p2 < nums2.size())` (2): `nums1[p1] < nums2[p2]`
 (D) (1): `ans == -1 && p1 < nums1.size() && p2 < nums2.size()` (2): `nums1[p1] > nums2[p2]`

Câu 41. [L.O.2.3] Dòng nào trong đoạn code dưới đây là KHÔNG hợp lệ:

```

int a = 10;
int b = 11;
const int* p = &a; // Line 3
*p = 100; // Line 4
p = &b; // Line 5

```

- (A) Tất cả đều hợp lệ (B) Dòng 3 (C) Dòng 5 (D) Dòng 4

Câu 42. [L.O.3.1] Khi xây dựng một lớp trong C++, nếu không xác định cụ thể, các thuộc tính và phương thức của lớp đó sẽ có tầm vực mặc định là:

- (A) package (B) protected (C) private (D) public

Câu 43. [L.O.1.1] Hãy cho biết output của mã giả sau:

```
Set numList to [4, 6, 0, 1, 8 ,3, 9]
Set res to numList[0]

For i from 1 to length of numList - 1:
    If numList[i] > res:
        Set res to numList[i-1]
End For

Output res
```

- (A) Các đáp án khác đều sai
(C) 9

- (B) 8
(D) 3

Câu 44. [L.O.2.1] Kết quả của fun (2) là gì?

```
int fun(int n) {
    if (n == 4)
        return n;
    else return 2*fun(n+1); }
```

(A) 8

(B) 4

(C) 16

(D) 32

Các thông tin sau sử dụng cho câu 45 đến 46:

Một bản đồ được biểu diễn bằng mảng 2 chiều gồm `n_rows` (một biến kiểu `int`) là số hàng và `n_cols` (một biến kiểu `int`) là số cột. Mỗi phần tử của mảng là một đối tượng thuộc class `MapElement`. Trên bản đồ có 2 loại phần tử là đường đi (biểu diễn bằng class `Path`) và tường (biểu diễn bằng class `Wall`). `Path` có thể cho phép một nhân vật di chuyển được trong khi tường thì không cho phép nhân vật di chuyển.

```
/* a */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    map[r] = new MapElement*[n_cols];
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Path();
    }
}
```

```
/* b */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    map[r] = new MapElement*[n_cols];
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}
```

```
/* c */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Path();
    }
}
```

```

/* d */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}

```

Câu 45. [L.O.2.3] Biến map được sử dụng để biểu diễn bản đồ. Giả sử mảng 2 chiều cho bản đồ được cấp phát động. Cho yêu cầu cần khởi tạo một bản đồ mà tất cả các phần tử đều là Path. Biết rằng Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu này:

- (A) Đoạn code bên dưới comment /* d */ (B) Đoạn code bên dưới comment /* a */
 (C) Đoạn code bên dưới comment /* b */ (D) Đoạn code bên dưới comment /* c */

Câu 46. [L.O.2.3] Biến map được sử dụng để biểu diễn bản đồ. Biết rằng: hiện tại bản đồ gồm tất cả các phần tử đều là Path; Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Cho yêu cầu: tạo ra một bản đồ gồm tất cả phần tử ở hàng đầu tiên đều là Wall. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu trên và không tạo ra rò rỉ bộ nhớ (memory leak):

```

/* a */
for (int i = 0; i < n_cols; ++i) {
    map[0][i] = new Wall();
}

```

```

/* b */
for (int i = 0; i < n_cols; ++i) {
    delete map[0][i];
    map[0][i] = new Wall();
}

```

```

/* c */
for (int i = 0; i < n_rows; ++i) {
    delete map[i][0];
    map[i][0] = new Wall();
}

```

```

/* d */
for (int i = 0; i < n_cols; ++i) {
    delete (*map[n_rows-1][i]);
    map[n_rows-1][i] = new Wall();
}

```

- (A) Đoạn code bên dưới comment /* d */ (B) Đoạn code bên dưới comment /* b */
 (C) Đoạn code bên dưới comment /* a */ (D) Đoạn code bên dưới comment /* c */

Các thông tin sau sử dụng cho câu 47. Một nhân vật có một túi đồ để chứa các vật phẩm có thể được thu thập trong khi di chuyển trên bản đồ. Class Character biểu diễn một nhân vật. Một Character có 2 thuộc tính private là hp và exp. Class Character cũng có các cặp hàm get, set để lấy và gán giá trị của thuộc tính tương ứng. Class BaseBag biểu diễn một túi đồ có mã nguồn được cho bên dưới. Abstract class BaseItem biểu diễn một vật phẩm. Túi đồ được hiện thực bằng danh sách liên kết, mỗi phần tử của danh sách là một Node, head và tail lần lượt là Node đầu tiên và Node cuối cùng của danh sách.

```

class Character {
private:
    int hp, exp;
public:
    int getHp() const;
    void setHp(int hp);
    int getExp() const;
    void setExp(int exp);
}

class BaseItem {
// Definition of class BaseItem
};

class Node {
friend class BaseBag;
private:
    BaseItem * item;
    Node * next;
public:
    Node(BaseItem * item, Node * next = NULL) {
        this->item = item;
        this->next = next;
    }
    ~Node() {}
};

```

```

class BaseBag {
private:
    Node * head, * tail;
    int capacity;
    int count;
protected:
    virtual void removeFirst() {
        if (count == 1) {
            delete head;
            head = NULL;
            tail = NULL;
        }
        else {
            /* r1 */
            /* r2 */
            /* r3 */
        }
        --count;
    }
public:
    BaseBag(int capacity) {
        this->capacity = capacity;
        this->count = 0;
    }
};

```

```

        this->head = this->tail = NULL;
    }
};

```

Câu 47. [L.O.2.3] Phương thức `removeFirst(Character * ch)` của `BaseBag` thực hiện xóa vật phẩm đầu tiên của túi đồ nếu túi đồ có ít nhất 1 vật phẩm (giả sử khi được gọi, phương thức luôn có ít nhất 1 vật phẩm). Chọn đoạn mã hợp lệ để điền vào các ghi chú (comment) `/* r1 */`, `/* r2 */`, `/* r3 */`:

- (A) `/* r1 */ Node * temp = head; /* r2 */ delete head; /* r3 */ head = temp->next;`
- (B) `/* r1 */ Node * temp = this->head; /* r2 */ head = head->next; /* r3 */ delete temp;`
- (C) `/* r1 */ delete head; /* r2 */ head = head->next; /* r3 */` (không cần điền mã)
- (D) `/* r1 */ head = head->next; /* r2 */ delete head; /* r3 */` (không cần điền mã)

Các thông tin sau sử dụng cho câu 48 đến 50:

Cho `struct PasswordInfo` mô tả thông tin về một mật khẩu trong mảng `arr_pwds[]` cụ thể như sau:

```

struct PasswordInfo {
    char pwd[50];
    int freq;
    int first_index;
};

```

Trong đó:

- `char pwd[50]`: Chuỗi password
- `int freq`: tần suất xuất hiện của password trong mảng `arr_pwds[]`
- `first_index`: Vị trí index xuất hiện đầu tiên của password

Hàm `void buildArrayUniquePassword>PasswordInfo arr_unique_pwds[], int & num_unique_pwds, const char * arr_pwds[], int num_pwds)` để tạo ra mảng các `PasswordInfo` không trùng nhau từ mảng các password trong `arr_pwds`.

Tham số đầu vào gồm:

- `PasswordInfo arr_unique_pwds[]`: Mảng các `PasswordInfo` không trùng nhau
- `int & num_unique_pwds`: Giá trị tham chiếu số lượng password không trùng nhau
- `arr_pwds` và `num_pwds` lần lượt là mảng các password và số lượng phần tử của mảng

Nội dung hàm như sau:

```

void buildArrayUniquePassword>PasswordInfo arr_unique_pwds[], int &
    num_unique_pwds, const char * arr_pwds[], int num_pwds) {
    bool first_meet;
    num_unique_pwds = 0;
    //First loop
    for (int i_f = 0; i_f < num_pwds; ++i_f) { // index_full
        first_meet = true;
        //Second loop
        for (int i_u = 0; i_u < num_unique_pwds; ++i_u) { // index_unique
            if (strcmp(arr_pwds[i_f], arr_unique_pwds[i_u].pwd) == 0) {
                ++arr_unique_pwds[i_u].freq;
                first_meet = false;
                break;
            }
        }
        //missing_stmt_a
        {
            strcpy(arr_unique_pwds[num_unique_pwds].pwd, arr_pwds[i_f]);

```



```

arr_unique_pwds[num_unique_pwds].first_index = i_f;
//missing_stmt_b
++num_unique_pwds;
    }
}
}

```

Câu 48. [L.O.2.2] Nếu thiếu câu lệnh `num_unique_pwds = 0` bên trong như hàm trên điều gì sẽ xảy ra?

- Ⓐ Số lần lặp nhiều hơn nhưng kết quả vẫn chính xác
- Ⓑ Chương trình có thể sẽ gây ra lỗi
- Ⓒ Chương trình sẽ chỉ lặp vòng lặp //Second loop qua đúng 1 lần tại phần tử đầu tiên của mảng `arr_unique`
- Ⓓ Giá trị `freq` của `PasswordInfo` tương ứng sẽ luôn luôn không được cập nhật

Câu 49. [L.O.2.2] Nội dung tại `//missing_stmt_a` có thể là:

- Ⓐ `if (!first_meet || !num_unique_pwds)`
- Ⓑ `if (first_meet)`
- Ⓒ `if(!first_meet)`
- Ⓓ `if (first_meet || num_unique_pwds)`

Câu 50. [L.O.2.2] Với nội dung đã chọn cho `//misng_stmt_a`, vậy nội dung câu lệnh có thể ở `//missing_stmt_b` là gì

- Ⓐ Đáp án A và B đều đúng
- Ⓑ `arr_unique_pwds[num_unique_pwds].freq = 1;`
- Ⓒ `arr_unique_pwds[num_unique_pwds].freq ++;`
- Ⓓ `arr_unique_pwds[num_unique_pwds].freq = -1;`

===== Kết thúc đề thi =====

Câu 1. D

Câu 2. B

Câu 3. C

Câu 4. A

Câu 5. A

Câu 6. C

Câu 7. C

Câu 8. A

Câu 9. B

Câu 10. D

Câu 11. C

Câu 12. A

Câu 13. C

Câu 14. A

Câu 15. B

Câu 16. C

Câu 17. B

Câu 18. D

Câu 19. B

Câu 20. B

Câu 21. D

Câu 22. D

Câu 23. C

Câu 24. B

Câu 25. C

Câu 26. A

Câu 27. A

Câu 28. C

Câu 29. A

Câu 30. A

Câu 31. B

Câu 32. C

Câu 33. C

Câu 34. B

Câu 35. C

Câu 36. C

Câu 37. D

Câu 38. B

Câu 39. D

Câu 40. B

Câu 41. D

Câu 42. C

Câu 43. D

Câu 44. C

Câu 45. B

Câu 46. B


Câu 47. B

Câu 48. B

Câu 49. B

Câu 50. B

Giảng viên ra đề: (Chữ ký và Họ tên)	(Ngày ra đề)	Người phê duyệt: (Chữ ký và họ tên)	(Ngày duyệt đề)
--	--------------	---	-----------------

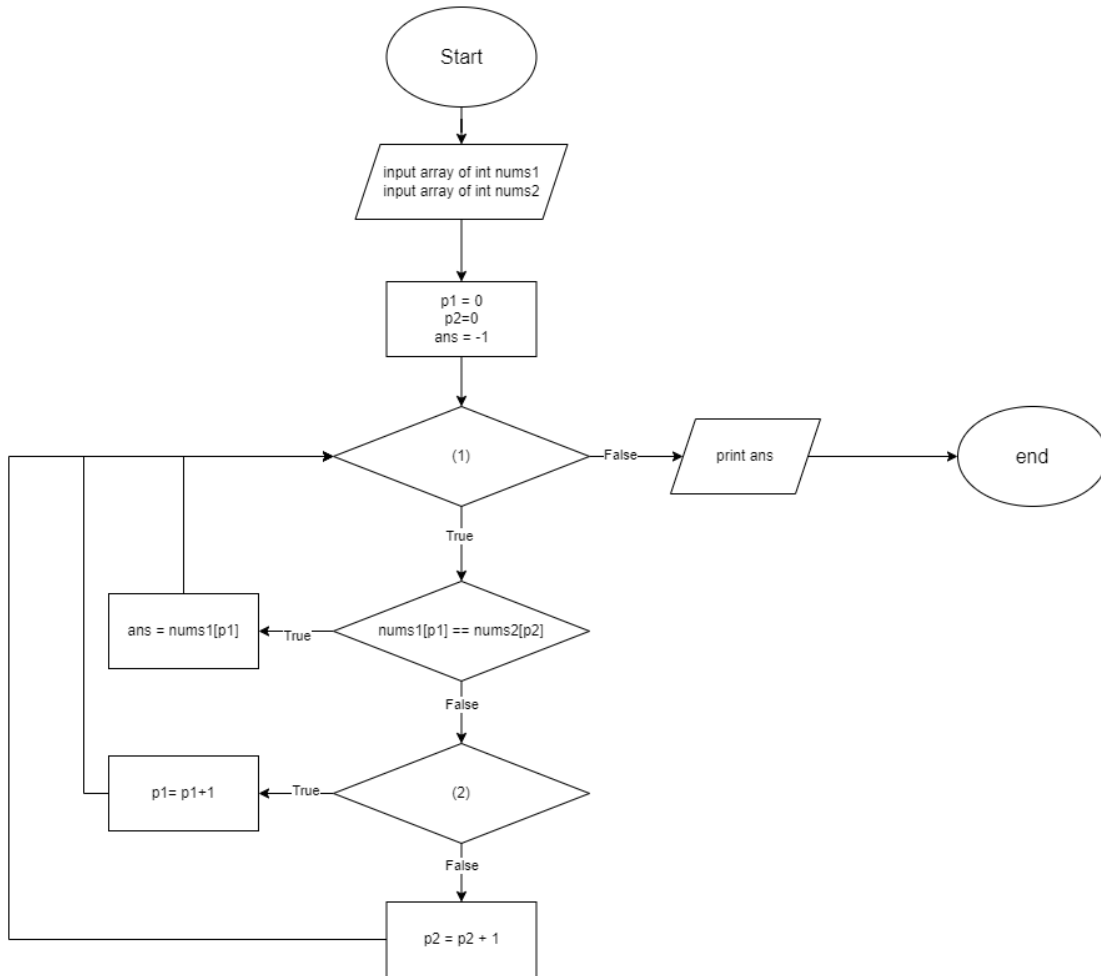
<div></div> <div>TRƯỜNG ĐH BÁCH KHOA - ĐHQG-HCM</div> <div><u>KHOA KH & KT MÁY TÍNH</u></div>	THI CUỐI KỲ		Học kỳ / Năm học	2	2023-2024
			Ngày thi	22-05-2024	
	Môn học	Kỹ thuật lập trình			
	Mã môn học	CO1027			
	Thời lượng	90 phút	Mã đề	2352	
Ghi chú: <ul style="list-style-type: none">- Đề thi có tổng cộng 60 câu.- Sinh viên được phép sử dụng một tờ A4 tài liệu tham khảo viết tay và máy tính cầm tay.- Các thư viện cơ bản đã được include đầy đủ.- Nộp lại đề sau khi kiểm tra/thi.					

Câu 1. [L.O.2.3] Dòng nào trong đoạn code dưới đây là KHÔNG hợp lệ:

```
int a = 10;
int b = 11;
const int* p = &a; // Line 3
*p = 100; // Line 4
p = &b; // Line 5
```

- (A) Dòng 3 (B) Tất cả đều hợp lệ (C) Dòng 5 (D) Dòng 4

Câu 2. [L.O.1.2] Để thực hiện thuật toán tìm phần tử nhỏ nhất *cùng xuất hiện* trong cả 2 mảng được sắp xếp theo thứ tự *tăng dần*, một thuật toán được thiết kế theo flowchart sau, hãy điền vào vị trí (1) và (2) sao cho chương trình sẽ in ra giá trị phần tử thỏa đề nếu có và ngược lại in -1 (Lưu ý, `nums1.size()` là lấy số phần tử có trong mảng `nums1`)



- (A) (1): `ans == -1 && p1 < nums1.size() && p2 < nums2.size()` (2): `nums1[p1] < nums2[p2]`
 (B) (1): `ans == -1 || (p1 < nums1.size() && p2 < nums2.size())` (2): `nums1[p1] > nums2[p2]`
 (C) (1): `ans == -1 || (p1 < nums1.size() && p2 < nums2.size())` (2): `nums1[p1] < nums2[p2]`
 (D) (1): `ans == -1 && p1 < nums1.size() && p2 < nums2.size()` (2): `nums1[p1] > nums2[p2]`

Câu 3. [L.O.3.1] Cho đoạn code dưới đây:

```
class Node {  
public:  
    Node() { cout << "Normal Constructor called\n"; }  
    Node(const Node &t) { cout << "Copy constructor called\n"; }  
};  
  
int main() {  
    Node *t1, *t2; // Line 1  
    t1 = new Node(); // Line 2  
    t2 = new Node(*t1); // Line 3  
    Node t3 = *t1; // Line 4  
    return 0;  
}
```

Chọn câu SAI:

- (A) Normal constructor được gọi ở dòng 2 (B) Copy constructor được gọi ở dòng 4
(C) Normal constructor được gọi ở dòng 1 (D) Copy constructor được gọi ở dòng 3

Câu 4. [L.O.3.1] Một đối tượng trong lập trình hướng đối tượng bao gồm:

- (A) Hằng số và biến (B) Thuộc tính và phương thức
(C) Cấu trúc dữ liệu (D) Toán tử và hàm

Câu 5. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main() {  
    int a = 1, b = 10, c = 3;  
    int *arr[3] = {&a, &b, &c};  
    cout << *arr[*arr[0] * 2] - c;  
    return 0;  
}
```

- (A) 3 (B) Lỗi biên dịch (C) 0 (D) Giá trị rác

Câu 6. [L.O.2.3] Cách khai báo biến tham chiếu ref (reference variable) trong C++ là gì?

- (A) ref *int; (B) int &ref; (C) ref int; (D) int *ref;

Câu 7. [L.O.2.1] Kết quả của fun (2) là gì?

```
int fun(int n) {  
    if (n == 4)  
        return n;  
    else return 2*fun(n+1); }
```

- (A) 4 (B) 8 (C) 16 (D) 32

Câu 8. [L.O.2.2] Trong C++, cho các string s1, s2 và s3 được khai báo như sau:

```
string s1 = "Hello_World";  
string s2(s1, 3);  
string s3(s2, 1, 4);
```

Giá trị của string s3 là?

- (A) o_Wo (B) _Wor (C) orld (D) ello

Câu 12. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int array1[] = {1, 2, 3, 4, 5};
int array2[] = {6, 7, 8, 9, 10};
int temp, result = 0;
int main() {
    for (temp = 0; temp < 5; temp++) {
        result += array1[temp];
    }
    for (; temp < 4; temp++) {
        result += array2[temp];
    }
    cout << result;
    return 0;
}
```

- (A) 15
(C) 45

- (B) 55
(D) 35

Câu 13. [L.O.2.4] Kết quả chương trình sau là gì?

```
#define foo(a, b) ((a) > (b) ? (a) : (b))
int main() {
    int x = 10, y = 20;
    cout << foo(x, y) << endl;
    return 0;
}
```

- (A) Lỗi biên dịch

- (B) Lỗi runtime

- (C) 10

- (D) 20

Câu 14. [L.O.1.1] Chọn phát biểu đúng về mô tả giải thuật:

- (A) Cần phải dùng mã lập trình

- (C) Cần phải dùng lưu đồ (flowchart)

- (B) Có nhiều cách khác nhau để mô tả giải thuật

- (D) Cần phải dùng mã giả (pseudocode)

Thông tin sau áp dụng cho câu hỏi 15:

```
int i = 0;

class A {
public:
    A() {
        i = 3;
    }
    ~A() {
        i=10;
    }
};

int foo() {
    int &p = i;
    A ob;
    return p;
}
```


Câu 15. [L.O.3.2] Lời gọi hàm *foo()* sẽ trả về giá trị nào?

- (A) 0 (B) Phương án khác (C) 3 (D) 10

Câu 16. [L.O.2.3] Hãy cho biết kết quả xuất ra màn hình của đoạn chương trình sau:

```
int x=1;
int *p=&x;
int &q=x;
cout << (q == *p);
```

- (A) 0 (B) Lỗi biên dịch (C) 1
(D) Địa chỉ của x và giá trị của x

Câu 17. [L.O.2.3] Kết quả của đoạn code C bên dưới là gì

```
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;
    printf("%d", *(ptr + 3));
    return 0;
}
```

- (A) 2 (B) Lỗi runtime (C) 5 (D) 4

Câu 18. [L.O.1.2] Ký hiệu nào dưới đây trong lưu đồ (flowchart) được dùng để biểu thị output được trả về?

- (A) Hình bình hành (B) Hình thoi
(C) Hình chữ nhật (D) Hình tròn/ellipse

Câu 19. [L.O.2.1] Độ phức tạp của hàm đệ quy sau là:

```
int recursiveFunction(int n) {
    if (n <= 1) {
        return 1;
    }
    return recursiveFunction(n - 1) + recursiveFunction(n - 2);
}
```

- (A) $O(n!)$ (B) $O(\log n)$ (C) $O(n^2)$ (D) $O(2^n)$

Câu 20. Biểu tượng hình thoi trong sơ đồ luồng (flowchart) biểu diễn cho:

- (A) Một điểm nơi mà có một câu hỏi được đặt và luồng được phân nhánh dựa vào câu trả lời (B) Điểm bắt đầu hoặc điểm kết thúc của luồng đó
(C) Một liên kết để kết nối các phần khác nhau của sơ đồ luồng (D) Một quy trình hoặc một hành động cụ thể

Câu 21. [L.O.2.1] Cho hiện thực hóa 1 hàm như sau:

```
void func(int* const list, int arraySize) {  
    for (int i=0; i<arraySize; i++)  
        list[i] = list[i] + 1;  
}  
int main() {  
    int numbers[5] = {5, 2, 7, 7, 1};  
    func(numbers, 5);  
    for (int i=0; i < 5; i++)  
        cout << numbers[i];  
    return 0;  
}
```

The value of array *numbers* when running the *main()* function is:

- (A) {6, 3, 8, 8, 2} (B) {5, 2, 7, 7, 1} (C) Lỗi khi chạy (D) Lỗi biên dịch

Câu 22. [L.O.3.1] Chọn phát biểu SAI trong các phát biểu sau:

- (A) Một class chỉ có thể có duy nhất một hàm hủy (destructor) (B) Hàm khởi tạo có thể là một hàm ảo (virtual function) trong abstract class
(C) Một class có thể có nhiều hàm khởi tạo (constructor) (D) Hàm khởi tạo có tham số truyền vào, nhưng hàm hủy thì không

Câu 23. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main(int argc, char const *argv[]) {  
    string str;  
    cin>>str;  
    cout<<str;  
    return 0;  
}
```

- (A) Chuỗi nhập vào bởi người dùng (B) str
(C) Error (Lỗi) (D) Garbage value (giá trị rác)

Câu 24. [L.O.3.1] Khi xây dựng một lớp trong C++, nếu không xác định cụ thể, các thuộc tính và phương thức của lớp đó sẽ có tầm vực mặc định là:

- (A) protected (B) package (C) private (D) public

Câu 25. [L.O.3.2] Thế nào được gọi là hiện tượng nạp chồng (overriding)?

- (A) Hiện tượng lớp con kế thừa định nghĩa một hàm hoàn toàn giống lớp cha
(B) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng các đối số nhưng khác kiểu với một hàm ở lớp cha
(C) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên nhưng khác kiểu với một hàm ở lớp cha
(D) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng kiểu với một hàm ở lớp cha nhưng khác đối số

Câu 26. [L.O.3.2] Trong lập trình hướng đối tượng (OOP), encapsulation có ý nghĩa gì?

- (A) Quy tắc cho phép một đối tượng che dấu thông tin bên trong và chỉ cho phép truy cập thông qua các phương thức
(B) Quy tắc cho phép một đối tượng tự quản lý bộ nhớ
(C) Quy tắc cho phép một đối tượng kế thừa các thuộc tính và phương thức từ một đối tượng khác
(D) Quy tắc cho phép một đối tượng kết nối với các đối tượng khác

Câu 27. [L.O.2.2] Kết quả in ra của đoạn chương trình sau, nếu chuỗi nhập vào ở lệnh cin là “I love BachKhoa”:

```
string st;  
cin >> st;  
cout << st;
```

- (A) I love BachKhoa (B) I (C) I love (D) BachKhoa

Câu 28. [L.O.2.3] Cách truy xuất vào một thành phần của đối tượng con trở this?

- (A) this_x (B) this.x (C) (*this)->x (D) this->x

Thông tin sau áp dụng cho câu hỏi 29:

```
class Triangle {  
public:  
    static int numOfTriangle;  
private:  
    int a;  
    int b;  
    int c;  
public:  
    Triangle(int a, int b, int c) : a(a), b(b), c(c) {  
        numOfTriangle++;  
    }  
    ~Triangle() {  
        numOfTriangle--;  
        cout << 3;  
    }  
};  
  
int Triangle::numOfTriangle = 0;  
int main() {  
    Triangle aList[5];  
    cout << (*aList).numOfTriangle;  
    return 0;  
}
```

Câu 29. [L.O.3.1] Cho biết kết quả in ra màn hình của chương trình:

- (A) 5 (B) Lỗi biên dịch (C) 53 (D) 533333

Câu 30. [L.O.1.1] Hãy cho biết output của mã giả sau:

```
Set matrix to [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Set finalMatrix to empty list

For i from 0 to 2:
    Set newRow to empty list
    For j from 0 to 2:
        Append matrix[2-j][i] to newRow
    End For
    Append newRow to finalMatrix
End For
```

- (A) [[9, 6, 3], [8, 5, 2], [7, 4, 1]] (B) [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
(C) [[1, 2, 3], [4, 5, 6], [7, 8, 9]] (D) [[3, 6, 9], [2, 5, 8], [1, 4, 7]]

Câu 31. [L.O.2.2] Phát biểu nào sau đây là ĐÚNG về Struct trong C++

- (A) Tầm vực truy xuất mặc định của các phần tử trong struct là private
(B) Struct không thể có phương thức (C) Struct không hỗ trợ kế thừa
(D) Có thể thay đổi tầm vực truy xuất của các thành phần trong struct

Câu 32. [L.O.2.3] Trong lập trình C, con trỏ **void*** có thể đại diện cho cái gì?

- (A) Con trỏ đến địa chỉ của một biến kiểu void (B) Con trỏ đến địa chỉ của kiểu dữ liệu bất kỳ
(C) Con trỏ đến một hàm có kiểu trả về là void (D) Con trỏ đến một biến hằng

Câu 33. [L.O.2.3] Câu nào mô tả đúng nhất về memory leak?

- (A) Chương trình cố tình phân bổ nhiều bộ nhớ hơn mức cần thiết.
(B) Chương trình không giải phóng được bộ nhớ không sử dụng, khiến bộ nhớ khả dụng dần cạn kiệt.
(C) Chương trình sử dụng bộ nhớ không hiệu quả, dẫn đến hiệu suất chậm hơn.
(D) Lỗi lập trình dẫn đến hỏng dữ liệu được lưu trữ trong bộ nhớ.

Cho chương trình như sau; giả sử rằng các thư viện cần thiết đã được **include**. Dùng đoạn mã này cho các câu hỏi từ 34 - 35.

```
/*Mark-1*/ A {
    A(int value=9): value_(value){ cout << "A, "; }
    ~A(){ cout << "~A, "; }
/*Mark-2*/
    int value_;
};
int main() {
    cout << "[";
    {A a; {A a; cout << "x, ";} cout << a.value_ << ", "; }
    cout << "]" ; return 0;
}
```

Câu 34. Thay thế **/*Mark-1*/** bởi **struct** và **/*Mark-2*/** bởi **public:**. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- (A) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private. (B) Có lỗi trong **main**, vì có hai biến cùng tên là **a**.
(C) Kết quả in ra là: [A, A, x] (D) Tất cả các lựa chọn khác đều sai.

Câu 35. Thay thế /*Mark-1*/ bởi **class** và /*Mark-2*/ bởi **public**:. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- (A) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private. (B) Có lỗi trong **main**, vì có hai biến cùng tên là **a**.
(C) Kết quả in ra là: [A, A, x, ~A, 9, ~A,] (D) Tất cả các lựa chọn khác đều sai.

Câu 36. [L.O.2.2] Trong ngôn ngữ lập trình C++, để khai báo một mảng có kích thước cố định và lưu trữ các số nguyên, ta sử dụng cú pháp nào?

- (A) `array<int> myArray(10);` (B) `int[] myArray = new int[10];`
(C) `int myArray[10];` (D) `array<int, 10> myArray;`

Câu 37. [L.O.3.1] Câu nào sau đây mô tả chính xác hàm ảo trong lập trình hướng đối tượng?

- (A) Một hàm có thể được ghi đè bởi (các) lớp kế thừa để cung cấp cách triển khai cụ thể.
(B) Một hàm có thể được gọi mà không cần một thể hiện đối tượng.
(C) Một hàm được khai báo bằng từ khóa `virtual` chỉ có thể được truy cập trong lớp nơi nó được khai báo.
(D) Hàm được gọi tự động khi một đối tượng được tạo.

Câu 38. [L.O.2.3] Trong đoạn code C++ dưới đây, giá trị nào sẽ được in ra sau khi chạy chương trình?

```
int main() {  
    char str[] = "hcmut-cse";  
    char *ptr = str;  
    ptr += 5;  
    cout << ptr;  
    return 0;  
}
```

- (A) -cse (B) t-cse
(C) cse (D) Chương trình bị lỗi biên dịch (Compilation Error)

Câu 39. [L.O.2.4] Kết quả khi chạy đoạn chương trình sau là:

```
int A[] = {2, 4, 1, 6, 8, 3, 1, 10};  
int *p = A;  
while (p < A + 6) {  
    cout << --*p++ << *p++;  
    p++;  
}
```

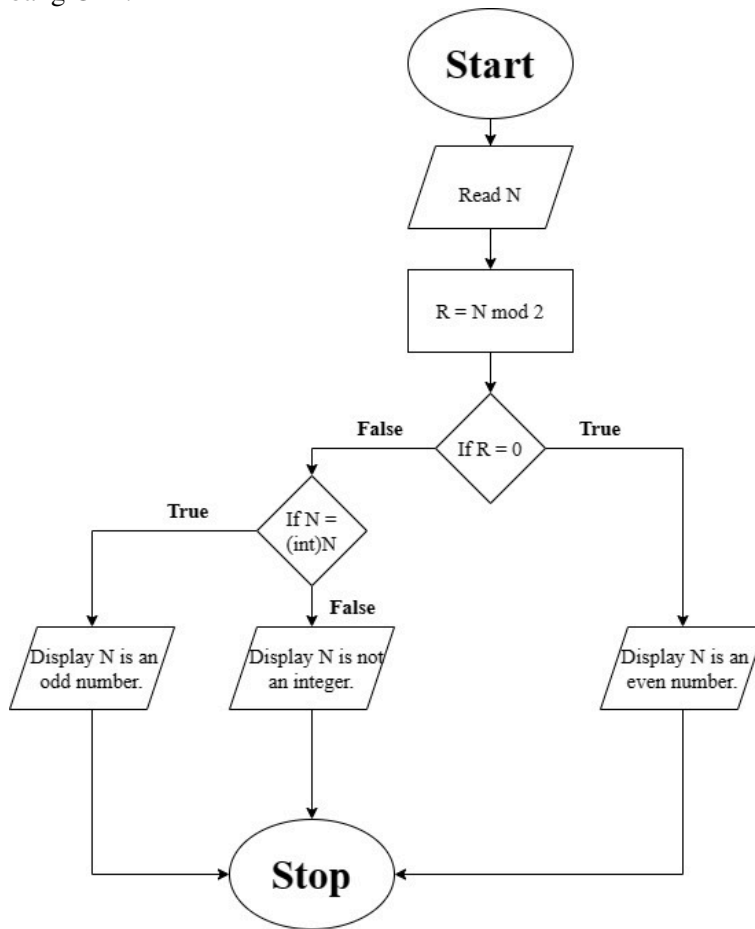
- (A) 1458 (B) 2367 (C) 4168 (D) 3158

Câu 40. [L.O.3.2] Class B sử dụng được các biến thành viên nào của class A:

```
class A {  
    int a, b;  
    protected:  
    double c, d;  
};  
class B : public A {  
};
```

- (A) a, b (B) Không truy cập được biến nào
(C) c, d (D) a, b, c, d

Câu 41. [L.O.1.2] Trường hợp nào không thể xảy ra với lưu đồ được cho bên dưới. Giả sử lưu đồ sẽ được hiện thực bằng C++.



- (A) Đầu vào: $N = 4$, Đầu ra: N is an even number
 (B) Đầu vào: $N = 1$, Đầu ra: N is an odd number
 (C) Đầu vào: $N = 3.5$, Đầu ra: N is not an integer
 (D) Tất cả các câu trên đều có thể xảy ra.

Câu 42. [L.O.3.2] Cho lớp A là cha của lớp B, và biến a có kiểu A, biến b có kiểu B.

Phát biểu gán 1: $a = b$;

Phát biểu gán 2: $b = a$;

- (A) Phát biểu 1 đúng, 2 sai
 (B) Cả hai phát biểu gán đều đúng
 (C) Phát biểu 1 sai, 2 đúng
 (D) Cả hai phát biểu gán đều đúng

Câu 43. [L.O.2.2] Trong C++, ta khai báo một mảng số nguyên hai chiều như sau: `int arr[5][7];`

Kích thước của một giá trị số nguyên là 4 byte. Địa chỉ ô nhớ bắt đầu của arr là 0×70 , địa chỉ vùng nhớ của `arr[3][4]` là bao nhiêu?

- (A) $0 \times a0$
 (B) $0 \times 8c$
 (C) $0 \times d4$
 (D) $0 \times fc$

Câu 44. [L.O.2.3] Mục đích của con trỏ trong lập trình là gì

- (A) Lưu trữ địa chỉ của biến
 (B) Cấp phát bộ nhớ động
 (C) Lưu trữ giá trị của biến
 (D) Thực hiện các phép tính số học

Các thông tin sau sử dụng cho câu 45. Một nhân vật có một túi đồ để chứa các vật phẩm có thể được thu thập trong khi di chuyển trên bản đồ. Class Character biểu diễn một nhân vật. Một Character có 2 thuộc tính private là hp và exp. Class Character cũng có các cặp hàm get, set để lấy và gán giá trị của thuộc tính tương ứng. Class BaseBag biểu diễn một túi đồ có mã nguồn được cho bên dưới. Abstract class BaseItem biểu diễn một vật phẩm. Túi đồ được hiện thực bằng danh sách liên kết, mỗi phần tử của danh sách là một Node, head và tail lần lượt là Node đầu tiên và Node cuối cùng của danh sách.

```
class Character {
```

```

private:
    int hp, exp;
public:
    int getHp() const;
    void setHp(int hp);
    int getExp() const;
    void setExp(int exp);
}

class BaseItem {
// Definition of class BaseItem
};

class Node {
friend class BaseBag;
private:
    BaseItem * item;
    Node * next;
public:
    Node(BaseItem * item, Node * next = NULL) {
        this->item = item;
        this->next = next;
    }
    ~Node() {}
};

```

```

class BaseBag {
private:
    Node * head, * tail;
    int capacity;
    int count;
protected:
    virtual void removeFirst() {
        if (count == 1) {
            delete head;
            head = NULL;
            tail = NULL;
        }
        else {
            /* r1 */
            /* r2 */
            /* r3 */
        }
        --count;
    }
public:
    BaseBag(int capacity) {
        this->capacity = capacity;
        this->count = 0;
        this->head = this->tail = NULL;
    }
};

```

```

    }
};

```

Câu 45. [L.O.2.3] Phương thức `removeFirst(Character * ch)` của `BaseBag` thực hiện xóa vật phẩm đầu tiên của túi đồ nếu túi đồ có ít nhất 1 vật phẩm (giả sử khi được gọi, phương thức luôn có ít nhất 1 vật phẩm). Chọn đoạn mã hợp lệ để điền vào các ghi chú (comment) `/* r1 */`, `/* r2 */`, `/* r3 */`:

- (A) `/* r1 */ Node * temp = this->head; /* r2 */ head = head->next; /* r3 */ delete temp;`
- (B) `/* r1 */ Node * temp = head; /* r2 */ delete head; /* r3 */ head = temp->next;`
- (C) `/* r1 */ delete head; /* r2 */ head = head->next; /* r3 */ (không cần điền mã)`
- (D) `/* r1 */ head = head->next; /* r2 */ delete head; /* r3 */ (không cần điền mã)`

Các thông tin sau sử dụng cho câu 46 đến 47:

Một bản đồ được biểu diễn bằng mảng 2 chiều gồm `n_rows` (một biến kiểu `int`) là số hàng và `n_cols` (một biến kiểu `int`) là số cột. Mỗi phần tử của mảng là một đối tượng thuộc class `MapElement`. Trên bản đồ có 2 loại phần tử là đường đi (biểu diễn bằng class `Path`) và tường (biểu diễn bằng class `Wall`). `Path` có thể cho phép một nhân vật di chuyển được trong khi tường thì không cho phép nhân vật di chuyển.

```

/* a */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    map[r] = new MapElement*[n_cols];
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Path();
    }
}

```

```

/* b */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    map[r] = new MapElement*[n_cols];
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}

```

```

/* c */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Path();
    }
}

```

```

/* d */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}

```


Câu 46. [L.O.2.3] Biến map được sử dụng để biểu diễn bản đồ. Giả sử mảng 2 chiều cho bản đồ được cấp phát động. Cho yêu cầu cần khởi tạo một bản đồ mà tất cả các phần tử đều là Path. Biết rằng Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu này:

- (A) Đoạn code bên dưới comment /* a */
 (B) Đoạn code bên dưới comment /* d */
 (C) Đoạn code bên dưới comment /* b */
 (D) Đoạn code bên dưới comment /* c */

Câu 47. [L.O.2.3] Biến map được sử dụng để biểu diễn bản đồ. Biết rằng: hiện tại bản đồ gồm tất cả các phần tử đều là Path; Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Cho yêu cầu: tạo ra một bản đồ gồm tất cả phần tử ở hàng đầu tiên đều là Wall. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu trên và không tạo ra rò rỉ bộ nhớ (memory leak):

```
/* a */
for (int i = 0; i < n_cols; ++i) {
    map[0][i] = new Wall();
}
```

```
/* b */
for (int i = 0; i < n_cols; ++i) {
    delete map[0][i];
    map[0][i] = new Wall();
}
```

```
/* c */
for (int i = 0; i < n_rows; ++i) {
    delete map[i][0];
    map[i][0] = new Wall();
}
```

```
/* d */
for (int i = 0; i < n_cols; ++i) {
    delete (*map[n_rows-1][i]);
    map[n_rows-1][i] = new Wall();
}
```

- (A) Đoạn code bên dưới comment /* b */
 (B) Đoạn code bên dưới comment /* d */
 (C) Đoạn code bên dưới comment /* a */
 (D) Đoạn code bên dưới comment /* c */

Các thông tin sau sử dụng cho câu 48 đến 50:

Cho struct PasswordInfo mô tả thông tin về một mật khẩu trong mảng arr_pwds[] cụ thể như sau:

```
struct PasswordInfo {
    char pwd[50];
    int freq;
    int first_index;
};
```

Trong đó:

- char pwd[50]: Chuỗi password
- int freq: tần suất xuất hiện của password trong mảng arr_pwds[]
- first_index: Vị trí index xuất hiện đầu tiên của password

Hàm void buildArrayUniquePassword(PasswordInfo arr_unique_pwds[], int & num_unique_pwds, const char * arr_pwds[], int num_pwds) để tạo ra mảng các PasswordInfo không trùng nhau từ mảng các password trong arr_pwds.

Tham số đầu vào gồm:

- PasswordInfo arr_unique_pwds[]: **Mảng** các PasswordInfo không trùng nhau
 - int & num_unique_pwds: **Giá trị** tham chiếu số lượng password không trùng nhau
 - arr_pwds và num_pwds lần lượt là mảng các password và số lượng phần tử của mảng
- Nội dung hàm như sau:

```
void buildArrayUniquePassword>PasswordInfo arr_unique_pwds[], int &
num_unique_pwds, const char * arr_pwds[], int num_pwds) {
    bool first_meet;
    num_unique_pwds = 0;
    //First loop
    for (int i_f = 0; i_f < num_pwds; ++i_f) { // index_full
        first_meet = true;
        //Second loop
        for (int iu = 0; iu < num_unique_pwds; ++iu) { // index_unique
            if (strcmp(arr_pwds[i_f], arr_unique_pwds[iu].pwd) == 0) {
                ++arr_unique_pwds[iu].freq;
                first_meet = false;
                break;
            }
        }
        //missing_stmt_a
        {
            strcpy(arr_unique_pwds[num_unique_pwds].pwd, arr_pwds[i_f]);
            arr_unique_pwds[num_unique_pwds].first_index = i_f;
            //missing_stmt_b
            ++num_unique_pwds;
        }
    }
}
```

Câu 48. [L.O.2.2] Nếu thiếu câu lệnh num_unique_pwds = 0 bên trong như hàm trên điều gì sẽ xảy ra?

- (A) Chương trình có thể sẽ gây ra lỗi
- (B) Số lần lặp nhiều hơn nhưng kết quả vẫn chính xác
- (C) Chương trình sẽ chỉ lặp vòng lặp //Second loop qua đúng 1 lần tại phần tử đầu tiên của mảng arr_unique
- (D) Giá trị freq của PasswordInfo tương ứng sẽ luôn luôn không được cập nhật

Câu 49. [L.O.2.2] Nội dung tại //missing_stmt_a có thể là:

- (A) if (first_meet)
- (B) if (!first_meet || !num_unique_pwds)
- (C) if(!first_meet)
- (D) if (first_meet || num_unique_pwds)

Câu 50. [L.O.2.2] Với nội dung đã chọn cho //misng_stmt_a, vậy nội dung câu lệnh có thể ở //missing_stmt_b là gì

- (A) arr_unique_pwds[num_unique_pwds].freq = 1;
- (B) Đáp án A và B đều đúng
- (C) arr_unique_pwds[num_unique_pwds].freq ++;
- (D) arr_unique_pwds[num_unique_pwds].freq = -1;

===== Kết thúc đề thi =====

- Câu 1.** D
Câu 2. A
Câu 3. C
Câu 4. B
Câu 5. C
Câu 6. B
Câu 7. C
Câu 8. A
Câu 9. B
Câu 10. C
Câu 11. D
Câu 12. A
Câu 13. D
Câu 14. B
- Câu 15.** C
- Câu 16.** C
Câu 17. D
Câu 18. A
Câu 19. D
Câu 20. A
Câu 21. A
Câu 22. B
Câu 23. A
Câu 24. C
Câu 25. A
Câu 26. A
Câu 27. B
Câu 28. D
- Câu 29.** B
- Câu 30.** A
Câu 31. D
Câu 32. B
Câu 33. B

Câu 34. D

Câu 35. A

Câu 36. C

Câu 37. A

Câu 38. A

Câu 39. A

Câu 40. C

Câu 41. D

Câu 42. A

Câu 43. C

Câu 44. A

Câu 45. A

Câu 46. A


Câu 47. A

Câu 48. A

Câu 49. A

Câu 50. A

Giảng viên ra đề: (Chữ ký và Họ tên)	(Ngày ra đề)	Người phê duyệt: (Chữ ký và họ tên)	(Ngày duyệt đề)
--	--------------	---	-----------------

<div></div> <div>TRƯỜNG ĐH BÁCH KHOA - ĐHQG-HCM</div> <div><u>KHOA KH & KT MÁY TÍNH</u></div>	THI CUỐI KỲ		Học kỳ / Năm học		2	2023-2024
			Ngày thi		22-05-2024	
	Môn học	Kỹ thuật lập trình				
	Mã môn học	CO1027				
	Thời lượng	90 phút	Mã đề	2353		
Ghi chú: <ul style="list-style-type: none">- Đề thi có tổng cộng 60 câu.- Sinh viên được phép sử dụng một tờ A4 tài liệu tham khảo viết tay và máy tính cầm tay.- Các thư viện cơ bản đã được include đầy đủ.- Nộp lại đề sau khi kiểm tra/thi.						

Câu 1. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int array1[] = {1, 2, 3, 4, 5};
int array2[] = {6, 7, 8, 9, 10};
int temp, result = 0;
int main() {
    for (temp = 0; temp < 5; temp++) {
        result += array1[temp];
    }
    for (; temp < 4; temp++) {
        result += array2[temp];
    }
    cout << result;
    return 0;
}
```

- ☐ (A) 15 ☐ (B) 45
☐ (C) 55 ☐ (D) 35

Câu 2. [L.O.2.1] Kết quả của fun (2) là gì?

```
int fun(int n) {
    if (n == 4)
        return n;
    else return 2*fun(n+1); }
```

- ☐ (A) 4 ☐ (B) 32 ☐ (C) 16 ☐ (D) 8

Câu 3. [L.O.3.1] Khi xây dựng một lớp trong C++, nếu không xác định cụ thể, các thuộc tính và phương thức của lớp đó sẽ có tầm vực mặc định là:

- ☐ (A) protected ☐ (B) public ☐ (C) private ☐ (D) package

Câu 4. [L.O.2.3] Cách truy xuất vào một thành phần của đối tượng con trở this?

- ☐ (A) this_x ☐ (B) this->x ☐ (C) (*this)->x ☐ (D) this.x

Câu 5. [L.O.2.3] Kết quả của đoạn code C bên dưới là gì

```
int main() {  
    int arr[] = {1, 2, 3, 4, 5};  
    int *ptr = arr;  
    printf("%d", *(ptr + 3));  
    return 0;  
}
```

- (A) 2 (B) 4 (C) 5 (D) Lỗi runtime

Câu 6. [L.O.1.1] Hãy cho biết output của mã giả sau:

```
Set matrix to [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
Set finalMatrix to empty list
```

```
For i from 0 to 2:  
    Set newRow to empty list  
    For j from 0 to 2:  
        Append matrix[2-j][i] to newRow  
    End For  
    Append newRow to finalMatrix  
End For
```

- (A) [[9, 6, 3], [8, 5, 2], [7, 4, 1]] (B) [[3, 6, 9], [2, 5, 8], [1, 4, 7]]
(C) [[1, 2, 3], [4, 5, 6], [7, 8, 9]] (D) [[1, 4, 7], [2, 5, 8], [3, 6, 9]]

Câu 7. [L.O.2.1] Cho đoạn mã chương trình sau của một hàm đệ qui và cách sử dụng hàm:

```
void recur_func(int n) {  
    if (n<=1)  
        return;  
    cout << n << " -> ";  
    if(n % 2)  
        recur_func(3*n+1);  
    else  
        recur_func(n/2);  
}  
int main() {  
    int n = 5;  
    recur_func(n);  
    return 0;  
}
```

Kết quả khi chạy hàm *main()* là:

- (A) 5 -> 2 -> 16 -> 2 -> 4 -> (B) 5 -> 4 -> 2 -> 16 -> 8 ->
(C) 5 -> 16 -> 8 -> 4 -> 2 -> (D) Runtime error (Lỗi chạy chương trình)

Câu 8. [L.O.2.2] Trong C++, ta khai báo một mảng số nguyên hai chiều như sau: **int** arr[5][7];
Kích thước của một giá trị số nguyên là 4 byte. Địa chỉ ô nhớ bắt đầu của arr là 0x70, địa chỉ vùng nhớ của arr[3][4] là bao nhiêu?

- (A) 0xa0 (B) 0xfc (C) 0xd4 (D) 0x8c

Câu 9. [L.O.2.3] Trong lập trình C, con trỏ **void*** có thể đại diện cho cái gì?

- (A) Con trỏ đến địa chỉ của một biến kiểu void (B) Con trỏ đến một biến hằng
(C) Con trỏ đến một hàm có kiểu trả về là void (D) Con trỏ đến địa chỉ của kiểu dữ liệu bất kỳ

Câu 10. [L.O.2.3] Trong đoạn code C++ dưới đây, giá trị nào sẽ được in ra sau khi chạy chương trình?

```
int main() {  
    char str[] = "hcmut-cse";  
    char *ptr = str;  
    ptr += 5;  
    cout << ptr;  
    return 0;  
}
```

- Ⓐ -cse
- Ⓒ t-cse

- Ⓑ cse
- Ⓓ Chương trình bị lỗi biên dịch (Compilation Error)

Câu 11. [L.O.3.2] Trong lập trình hướng đối tượng (OOP), encapsulation có ý nghĩa gì?

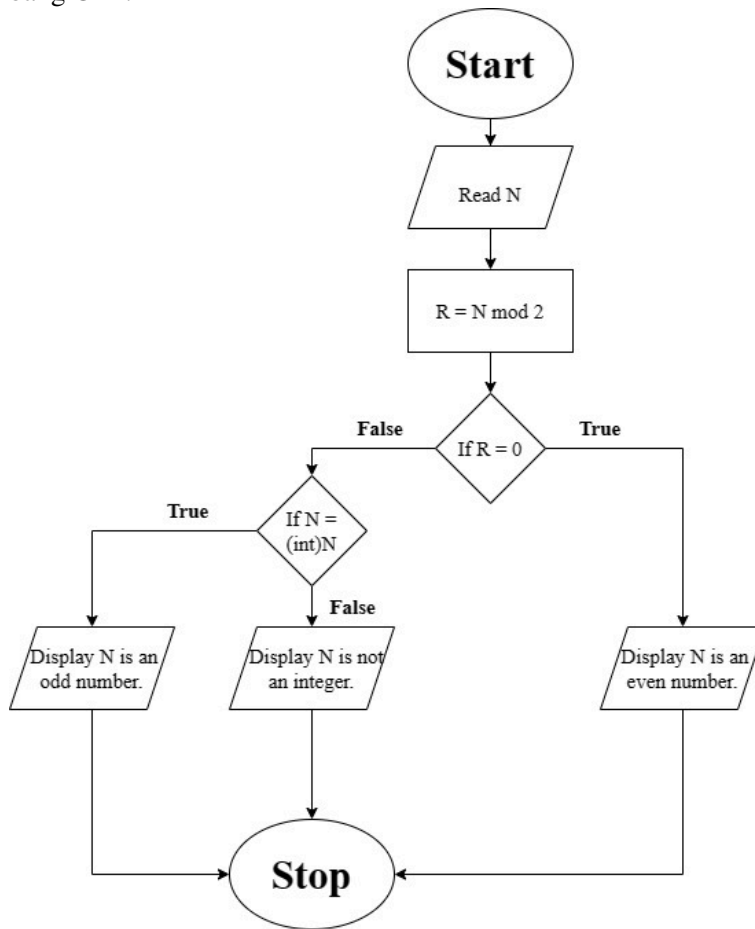
- Ⓐ Quy tắc cho phép một đối tượng che giấu thông tin bên trong và chỉ cho phép truy cập thông qua các phương thức
- Ⓑ Quy tắc cho phép một đối tượng kết nối với các đối tượng khác
- Ⓒ Quy tắc cho phép một đối tượng kế thừa các thuộc tính và phương thức từ một đối tượng khác
- Ⓓ Quy tắc cho phép một đối tượng tự quản lý bộ nhớ

Câu 12. [L.O.2.4] Kết quả khi chạy đoạn chương trình sau là:

```
int A[] = {2, 4, 1, 6, 8, 3, 1, 10};  
int *p = A;  
while (p < A + 6) {  
    cout << --*p++ << *p++;  
    p++;  
}
```

- Ⓐ 1458
- Ⓑ 3158
- Ⓒ 4168
- Ⓓ 2367

Câu 13. [L.O.1.2] Trường hợp nào không thể xảy ra với lưu đồ được cho bên dưới. Giả sử lưu đồ sẽ được hiện thực bằng C++.



- (A) Đầu vào: N = 4, Đầu ra: N is an even number (B) Đầu vào: N = 3.5, Đầu ra: N is not an integer
 (C) Đầu vào: N = 1, Đầu ra: N is an odd number (D) Tất cả các câu trên đều có thể xảy ra.

Câu 14. [L.O.2.3] Cách khai báo biến tham chiếu ref (reference variable) trong C++ là gì?

- (A) ref *int; (B) int *ref; (C) ref int; (D) int &ref;

Cho chương trình như sau; giả sử rằng các thư viện cần thiết đã được **include**. Dùng đoạn mã này cho các câu hỏi từ 15 - 16.

```

/*Mark-1*/ A {
    A(int value=9): value_(value){ cout << "A, "; }
    ~A(){ cout << "~A, "; }
/*Mark-2*/
    int value_;
};

int main() {
    cout << "[";
    {A a; {A a; cout << "x, ";} cout << a.value_ << ", "; }
    cout << "]" ; return 0;
}
  
```


Câu 15. Thay thế /*Mark-1*/ bởi **struct** và /*Mark-2*/ bởi **public**:. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- (A) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private. (B) Kết quả in ra là: [A, A, x]
(C) Có lỗi trong **main**, vì có hai biến cùng tên là **a**. (D) Tất cả các lựa chọn khác đều sai.

Câu 16. Thay thế /*Mark-1*/ bởi **class** và /*Mark-2*/ bởi **public**:. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- (A) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private. (B) Kết quả in ra là: [A, A, x, ~A, 9, ~A,]
(C) Có lỗi trong **main**, vì có hai biến cùng tên là **a**. (D) Tất cả các lựa chọn khác đều sai.

Câu 17. [L.O.2.2] Trong C++, cho các string s1, s2 và s3 được khai báo như sau:

```
string s1 = "Hello_World";  
string s2(s1, 3);  
string s3(s2, 1, 4);
```

Giá trị của string s3 là?

- (A) o_Wo (B) ello (C) orld (D) _Wor

Câu 18. [L.O.2.2] Trong ngôn ngữ lập trình C++, để khai báo một mảng có kích thước cố định và lưu trữ các số nguyên, ta sử dụng cú pháp nào?

- (A) `array<int> myArray(10);` (B) `array<int, 10> myArray;`
(C) `int myArray[10];` (D) `int[] myArray = new int[10];`

Câu 19. [L.O.3.1] Chọn phát biểu SAI trong các phát biểu sau:

- (A) Một class chỉ có thể có duy nhất một hàm hủy (destructor) (B) Hàm khởi tạo có tham số truyền vào, nhưng hàm hủy thì không
(C) Một class có thể có nhiều hàm khởi tạo (constructor) (D) Hàm khởi tạo có thể là một hàm ảo (virtual function) trong abstract class

Câu 20. [L.O.3.1] Một đối tượng trong lập trình hướng đối tượng bao gồm:

- (A) Hằng số và biến (B) Toán tử và hàm
(C) Cấu trúc dữ liệu (D) Thuộc tính và phương thức

Câu 21. [L.O.2.1] Cho hiện thực hóa 1 hàm như sau:

```
void func(int* const list, int arraySize) {  
    for (int i=0; i<arraySize; i++)  
        list[i] = list[i] + 1;  
}  
int main() {  
    int numbers[5] = {5, 2, 7, 7, 1};  
    func(numbers, 5);  
    for (int i=0; i < 5; i++)  
        cout << numbers[i];  
    return 0;  
}
```

The value of array *numbers* when running the *main()* function is:

- (A) {6, 3, 8, 8, 2} (B) Lỗi khi chạy (C) {5, 2, 7, 7, 1} (D) Lỗi biên dịch

Câu 22. [L.O.3.1] Câu nào sau đây mô tả chính xác hàm ảo trong lập trình hướng đối tượng?

- (A) Một hàm có thể được ghi đè bởi (các) lớp kế thừa để cung cấp cách triển khai cụ thể.
- (B) Hàm được gọi tự động khi một đối tượng được tạo.
- (C) Một hàm được khai báo bằng từ khóa virtual chỉ có thể được truy cập trong lớp nơi nó được khai báo.
- (D) Một hàm có thể được gọi mà không cần một thể hiện đối tượng.

Câu 23. [L.O.3.1] Điều nào sau đây mô tả đúng nhất các lớp trừu tượng?

- (A) Nếu một lớp có nhiều hơn một hàm ảo thì đó là lớp trừu tượng
- (B) Nếu một lớp chỉ có một hàm ảo thuần túy thì đó là lớp trừu tượng
- (C) Nếu một lớp có ít nhất một hàm ảo thuần túy thì đó là lớp trừu tượng
- (D) Nếu một lớp chỉ có tất cả các hàm ảo thuần túy thì đó là lớp trừu tượng

Câu 24. [L.O.3.2] Cho lớp A là cha của lớp B, và biến a có kiểu A, biến b có kiểu B.

Phát biểu gán 1: $a = b$;

Phát biểu gán 2: $b = a$;

- (A) Phát biểu 1 đúng, 2 sai
- (B) Cả hai phát biểu gán đều đúng
- (C) Phát biểu 1 sai, 2 đúng
- (D) Cả hai phát biểu gán đều đúng

Thông tin sau áp dụng cho câu hỏi 25:

```
int i = 0;

class A {
public:
    A() {
        i = 3;
    }
    ~A() {
        i=10;
    }
};

int foo() {
    int &p = i;
    A ob;
    return p;
}
```

Câu 25. [L.O.3.2] Lỗi gọi hàm *foo()* sẽ trả về giá trị nào?

- (A) 0
- (B) 10
- (C) 3
- (D) Phương án khác

Câu 26. Biểu tượng hình thoi trong sơ đồ luồng (flowchart) biểu diễn cho:

- (A) Một điểm nơi mà có một câu hỏi được đặt và luồng được phân nhánh dựa vào câu trả lời
- (B) Một quy trình hoặc một hành động cụ thể
- (C) Một liên kết để kết nối các phần khác nhau của sơ đồ luồng
- (D) Điểm bắt đầu hoặc điểm kết thúc của luồng đó

Câu 27. [L.O.1.1] Chọn phát biểu đúng về mô tả giải thuật:

- (A) Cần phải dùng mã lập trình
- (B) Cần phải dùng mã giả (pseudocode)
- (C) Cần phải dùng lưu đồ (flowchart)
- (D) Có nhiều cách khác nhau để mô tả giải thuật

Thông tin sau áp dụng cho câu hỏi 28:

```

class Triangle {
public:
    static int numOfTriangle;
private:
    int a;
    int b;
    int c;
public:
    Triangle(int a, int b, int c) : a(a), b(b), c(c) {
        numOfTriangle++;
    }
    ~Triangle() {
        numOfTriangle--;
        cout << 3;
    }
};

int Triangle::numOfTriangle = 0;
int main() {
    Triangle aList[5];
    cout << (*aList).numOfTriangle;
    return 0;
}

```

Câu 28. [L.O.3.1] Cho biết kết quả in ra màn hình của chương trình:

- ☐ (A) 5
 ☐ (B) 533333
 ☐ (C) 53
 ☐ (D) Lỗi biên dịch

Câu 29. [L.O.2.3] Mục đích của con trỏ trong lập trình là gì

- ☐ (A) Lưu trữ địa chỉ của biến
 ☐ (B) Thực hiện các phép tính số học
☐ (C) Lưu trữ giá trị của biến
 ☐ (D) Cấp phát bộ nhớ động

Câu 30. [L.O.2.2] Kết quả in ra của đoạn chương trình sau, nếu chuỗi nhập vào ở lệnh cin là "I love BachKhoa":

```

string st;
cin >> st;
cout << st;

```

- ☐ (A) I love BachKhoa
 ☐ (B) BachKhoa
 ☐ (C) I love
 ☐ (D) I

Câu 31. [L.O.3.2] Class B sử dụng được các biến thành viên nào của class A:

```

class A {
    int a, b;
protected:
    double c, d;
};
class B : public A {
};

```

- ☐ (A) a, b
 ☐ (B) a, b, c, d
☐ (C) c, d
 ☐ (D) Không truy cập được biến nào

Câu 32. [L.O.3.1] Cho đoạn code dưới đây:

```
class Node {  
public:  
    Node() { cout << "Normal Constructor called\n"; }  
    Node(const Node &t) { cout << "Copy constructor called\n"; }  
};  
  
int main() {  
    Node *t1, *t2; // Line 1  
    t1 = new Node(); // Line 2  
    t2 = new Node(*t1); // Line 3  
    Node t3 = *t1; // Line 4  
    return 0;  
}
```

Chọn câu SAI:

- (A) Normal constructor được gọi ở dòng 2
- (B) Copy constructor được gọi ở dòng 3
- (C) Normal constructor được gọi ở dòng 1
- (D) Copy constructor được gọi ở dòng 4

Câu 33. [L.O.3.2] Thế nào được gọi là hiện tượng nạp chồng (overriding)?

- (A) Hiện tượng lớp con kế thừa định nghĩa một hàm hoàn toàn giống lớp cha
- (B) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng kiểu với một hàm ở lớp cha nhưng khác đối số
- (C) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên nhưng khác kiểu với một hàm ở lớp cha
- (D) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng các đối số nhưng khác kiểu với một hàm ở lớp cha

Câu 34. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main(int argc, char const *argv[]) {  
    string str;  
    cin>>str;  
    cout<<str;  
    return 0;  
}
```

- (A) Chuỗi nhập vào bởi người dùng
- (B) Error (Lỗi)
- (C) str
- (D) Garbage value (giá trị rác)

Câu 35. [L.O.2.2] Phát biểu nào sau đây là ĐÚNG về Struct trong C++

- (A) Tầm vực truy xuất mặc định của các phần tử trong struct là private
- (B) Có thể thay đổi tầm vực truy xuất của các thành phần trong struct
- (C) Struct không hỗ trợ kế thừa
- (D) Struct không thể có phương thức

Câu 36. [L.O.1.2] Ký hiệu nào dưới đây trong lưu đồ (flowchart) được dùng để biểu thị output được trả về?

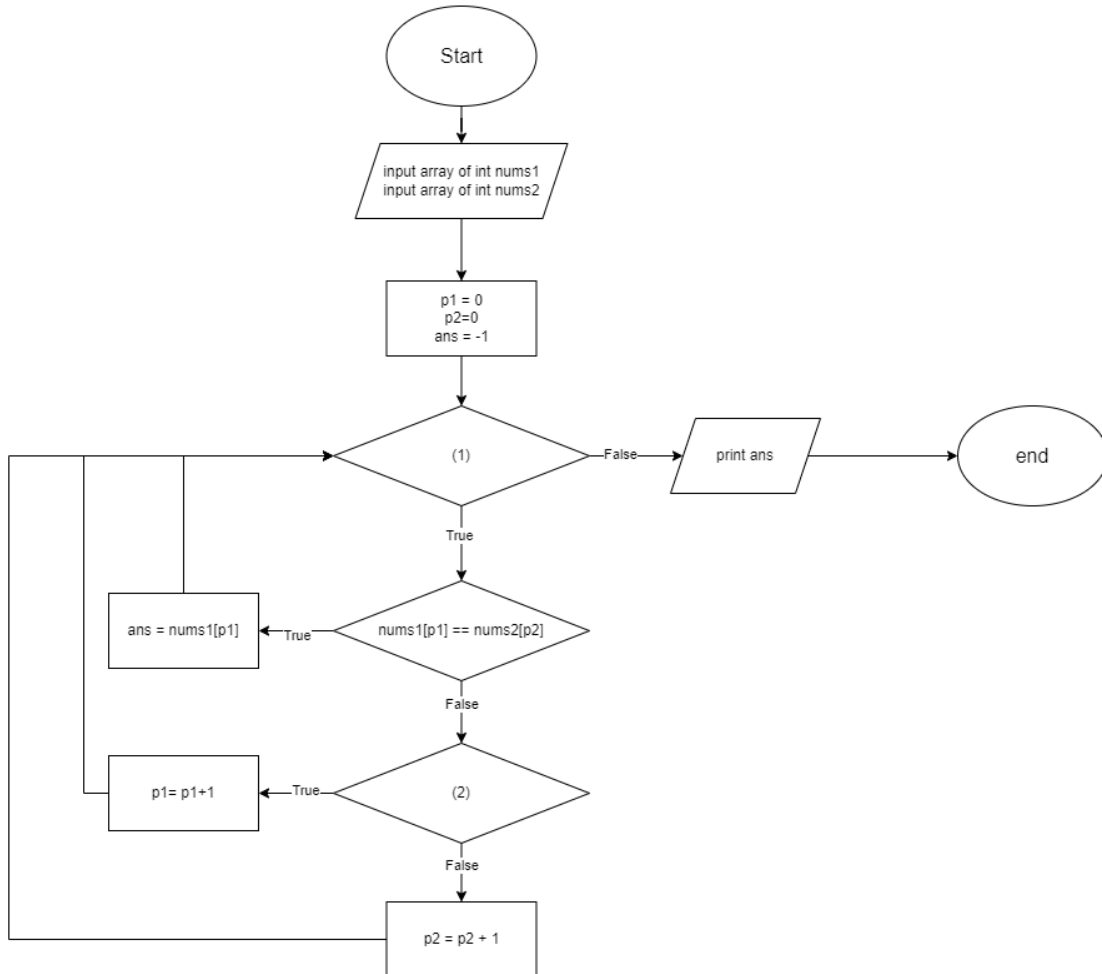
- (A) Hình bình hành
- (B) Hình tròn/ellipse
- (C) Hình chữ nhật
- (D) Hình thoi

Câu 37. [L.O.2.1] Độ phức tạp của hàm đệ quy sau là:

```
int recursiveFunction(int n) {
    if (n <= 1) {
        return 1;
    }
    return recursiveFunction(n - 1) + recursiveFunction(n - 2);
}
```

- (A) $O(n!)$
 (B) $O(2^n)$
 (C) $O(n^2)$
 (D) $O(\log n)$

Câu 38. [L.O.1.2] Để thực hiện thuật toán tìm phần tử nhỏ nhất **cùng xuất hiện** trong cả 2 mảng được sắp xếp theo thứ tự **tăng dần**, một thuật toán được thiết kế theo flowchart sau, hãy điền vào vị trí (1) và (2) sao cho chương trình sẽ in ra giá trị phần tử thỏa đề nếu có và ngược lại in -1 (Lưu ý, `nums1.size()` là lấy số phần tử có trong mảng `nums1`)



- (A) (1): `ans == -1 && p1 < nums1.size() && p2 < nums2.size()` (2): `nums1[p1] < nums2[p2]`
 (B) (1): `ans == -1 && p1 < nums1.size() && p2 < nums2.size()` (2): `nums1[p1] > nums2[p2]`
 (C) (1): `ans == -1 || (p1 < nums1.size() && p2 < nums2.size())` (2): `nums1[p1] < nums2[p2]`
 (D) (1): `ans == -1 || (p1 < nums1.size() && p2 < nums2.size())` (2): `nums1[p1] > nums2[p2]`

Câu 39. [L.O.2.4] Kết quả chương trình sau là gì?

```
#define foo(a, b) ((a) > (b) ? (a) : (b))
int main() {
    int x = 10, y = 20;
    cout << foo(x, y) << endl;
    return 0;
}
```

- (A) Lỗi biên dịch (B) 20 (C) 10 (D) Lỗi runtime

Câu 40. [L.O.2.3] Hãy cho biết kết quả xuất ra màn hình của đoạn chương trình sau:

```
int x=1;
int *p=&x;
int &q=x;
cout << (q == *p);
```

- (A) 0 (B) Địa chỉ của x và giá trị của x (C) 1
(D) Lỗi biên dịch

Câu 41. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main() {
    int a = 1, b = 10, c = 3;
    int *arr[3] = {&a, &b, &c};
    cout << *arr[*arr[0] * 2] - c;
    return 0;
}
```

- (A) 3 (B) Giá trị rác (C) 0 (D) Lỗi biên dịch

Câu 42. [L.O.1.1] Hãy cho biết output của mã giả sau:

```
Set numList to [4, 6, 0, 1, 8, 3, 9]
Set res to numList[0]

For i from 1 to length of numList - 1:
    If numList[i] > res:
        Set res to numList[i-1]
End For

Output res
```

- (A) 8 (B) 3
(C) 9 (D) Các đáp án khác đều sai

Câu 43. [L.O.2.3] Câu nào mô tả đúng nhất về memory leak?

- (A) Chương trình cố tình phân bổ nhiều bộ nhớ hơn mức cần thiết.
(B) Lỗi lập trình dẫn đến dữ liệu được lưu trữ trong bộ nhớ.
(C) Chương trình sử dụng bộ nhớ không hiệu quả, dẫn đến hiệu suất chậm hơn.
(D) Chương trình không giải phóng được bộ nhớ không sử dụng, khiến bộ nhớ khả dụng dần cạn kiệt.

Câu 44. [L.O.2.3] Dòng nào trong đoạn code dưới đây là KHÔNG hợp lệ:

```
int a = 10;
int b = 11;
const int* p = &a; // Line 3
*p = 100; // Line 4
p = &b; // Line 5
```

Ⓐ Dòng 3

Ⓑ Dòng 4

Ⓒ Dòng 5

Ⓓ Tất cả đều hợp lệ

Các thông tin sau sử dụng cho câu 45 đến 47:

Cho struct PasswordInfo mô tả thông tin về một mật khẩu trong mảng arr_pwds[] cụ thể như sau:

```
struct PasswordInfo {
    char pwd[50];
    int freq;
    int first_index;
};
```

Trong đó:

- char pwd[50]: Chuỗi password

- int freq: tần suất xuất hiện của password trong mảng arr_pwds[]

- first_index: Vị trí index xuất hiện đầu tiên của password

Hàm void buildArrayUniquePassword>PasswordInfo arr_unique_pwds[], int & num_unique_pwds, const char * arr_pwds[], int num_pwds) để tạo ra mảng các PasswordInfo không trùng nhau từ mảng các password trong arr_pwds.

Tham số đầu vào gồm:

- PasswordInfo arr_unique_pwds[]: Mảng các PasswordInfo không trùng nhau

- int & num_unique_pwds: Giá trị tham chiếu số lượng password không trùng nhau

- arr_pwds và num_pwds lần lượt là mảng các password và số lượng phần tử của mảng

Nội dung hàm như sau:

```
void buildArrayUniquePassword>PasswordInfo arr_unique_pwds[], int &
num_unique_pwds, const char * arr_pwds[], int num_pwds) {
    bool first_meet;
    num_unique_pwds = 0;
    //First loop
    for (int i_f = 0; i_f < num_pwds; ++i_f) { // index_full
        first_meet = true;
        //Second loop
        for (int i_u = 0; i_u < num_unique_pwds; ++i_u) { // index_unique
            if (strcmp(arr_pwds[i_f], arr_unique_pwds[i_u].pwd) == 0) {
                ++arr_unique_pwds[i_u].freq;
                first_meet = false;
                break;
            }
        }
        //missing_stmt_a
        {
            strcpy(arr_unique_pwds[num_unique_pwds].pwd, arr_pwds[i_f]);
            arr_unique_pwds[num_unique_pwds].first_index = i_f;
            //missing_stmt_b
            ++num_unique_pwds;
        }
    }
}
```

```

    }
}

```

Câu 45. [L.O.2.2] Nếu thiếu câu lệnh `num_unique_pwds = 0` bên trong như hàm trên điều gì sẽ xảy ra?

- (A) Chương trình có thể sẽ gây ra lỗi
- (B) Giá trị `freq` của `PasswordInfo` tương ứng sẽ luôn luôn không được cập nhật
- (C) Chương trình sẽ chỉ lặp vòng lặp `//Second loop` qua đúng 1 lần tại phần tử đầu tiên của mảng `arr_unique`
- (D) Số lần lặp nhiều hơn nhưng kết quả vẫn chính xác

Câu 46. [L.O.2.2] Nội dung tại `//missing_stmt_a` có thể là:

- (A) `if (first_meet)`
- (B) `if (first_meet || num_unique_pwds)`
- (C) `if(!first_meet)`
- (D) `if (!first_meet || !num_unique_pwds)`

Câu 47. [L.O.2.2] Với nội dung đã chọn cho `//misng_stmt_a`, vậy nội dung câu lệnh có thể ở `//missing_stmt_b` là gì

- (A) `arr_unique_pwds[num_unique_pwds].freq = 1;`
- (B) `arr_unique_pwds[num_unique_pwds].freq = -1;`
- (C) `arr_unique_pwds[num_unique_pwds].freq ++;`
- (D) Đáp án A và B đều đúng

Các thông tin sau sử dụng cho câu 48. Một nhân vật có một túi đồ để chứa các vật phẩm có thể được thu thập trong khi di chuyển trên bản đồ. Class `Character` biểu diễn một nhân vật. Một `Character` có 2 thuộc tính private là `hp` và `exp`. Class `Character` cũng có các cặp hàm `get`, `set` để lấy và gán giá trị của thuộc tính tương ứng. Class `BaseBag` biểu diễn một túi đồ có mã nguồn được cho bên dưới. Abstract class `BaseItem` biểu diễn một vật phẩm. Túi đồ được hiện thực bằng danh sách liên kết, mỗi phần tử của danh sách là một `Node`, `head` và `tail` lần lượt là `Node` đầu tiên và `Node` cuối cùng của danh sách.

```

class Character {
private:
    int hp, exp;
public:
    int getHp() const;
    void setHp(int hp);
    int getExp() const;
    void setExp(int exp);
}

class BaseItem {
// Definition of class BaseItem
};

class Node {
friend class BaseBag;
private:
    BaseItem * item;
    Node * next;
public:
    Node(BaseItem * item, Node * next = NULL) {
        this->item = item;
        this->next = next;
    }
}

```



```

    ~Node() {}
};

class BaseBag {
private:
    Node * head, * tail;
    int capacity;
    int count;
protected:
    virtual void removeFirst() {
        if (count == 1) {
            delete head;
            head = NULL;
            tail = NULL;
        }
        else {
            /* r1 */
            /* r2 */
            /* r3 */
        }
        --count;
    }
public:
    BaseBag(int capacity) {
        this->capacity = capacity;
        this->count = 0;
        this->head = this->tail = NULL;
    }
};

```

Câu 48. [L.O.2.3] Phương thức removeFirst(Character * ch) của BaseBag thực hiện xóa vật phẩm đầu tiên của túi đồ nếu túi đồ có ít nhất 1 vật phẩm (giả sử khi được gọi, phương thức luôn có ít nhất 1 vật phẩm). Chọn đoạn mã hợp lệ để điền vào các ghi chú (comment) /* r1 */, /* r2 */, /* r3 */:

- (A) /* r1 */ Node * temp = this->head; /* r2 */ head = head->next; /* r3 */ delete temp;
- (B) /* r1 */ head = head->next; /* r2 */ delete head; /* r3 */ (không cần điền mã)
- (C) /* r1 */ delete head; /* r2 */ head = head->next; /* r3 */ (không cần điền mã)
- (D) /* r1 */ Node * temp = head; /* r2 */ delete head; /* r3 */ head = temp->next;

Các thông tin sau sử dụng cho câu 49 đến 50:

Một bản đồ được biểu diễn bằng mảng 2 chiều gồm n_rows (một biến kiểu int) là số hàng và n_cols (một biến kiểu int) là số cột. Mỗi phần tử của mảng là một đối tượng thuộc class MapElement. Trên bản đồ có 2 loại phần tử là đường đi (biểu diễn bằng class Path) và tường (biểu diễn bằng class Wall). Path có thể cho phép một nhân vật di chuyển được trong khi tường thì không cho phép nhân vật di chuyển.

```

/* a */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    map[r] = new MapElement*[n_cols];
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Path();
    }
}

```

```

    }
}

/* b */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    map[r] = new MapElement*[n_cols];
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}

/* c */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Path();
    }
}

/* d */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}

```

Câu 49. [L.O.2.3] Biến map được sử dụng để biểu diễn bản đồ. Giả sử mảng 2 chiều cho bản đồ được cấp phát động. Cho yêu cầu cần khởi tạo một bản đồ mà tất cả các phần tử đều là Path. Biết rằng Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu này:

- | | |
|--|--|
| (A) Đoạn code bên dưới comment /* a */ | (B) Đoạn code bên dưới comment /* c */ |
| (C) Đoạn code bên dưới comment /* b */ | (D) Đoạn code bên dưới comment /* d */ |

Câu 50. [L.O.2.3] Biến map được sử dụng để biểu diễn bản đồ. Biết rằng: hiện tại bản đồ gồm tất cả các phần tử đều là Path; Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Cho yêu cầu: tạo ra một bản đồ gồm tất cả phần tử ở hàng đầu tiên đều là Wall. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu trên và không tạo ra rò rỉ bộ nhớ (memory leak):

```
/* a */
for (int i = 0; i < n_cols; ++i) {
    map[0][i] = new Wall();
}

/* b */
for (int i = 0; i < n_cols; ++i) {
    delete map[0][i];
    map[0][i] = new Wall();
}

/* c */
for (int i = 0; i < n_rows; ++i) {
    delete map[i][0];
    map[i][0] = new Wall();
}

/* d */
for (int i = 0; i < n_cols; ++i) {
    delete (*map[n_rows-1][i]);
    map[n_rows-1][i] = new Wall();
}
```

(A) Đoạn code bên dưới comment /* b */

(B) Đoạn code bên dưới comment /* c */

(C) Đoạn code bên dưới comment /* a */

(D) Đoạn code bên dưới comment /* d */

===== Kết thúc đề thi =====

Câu 1. A
Câu 2. C
Câu 3. C
Câu 4. B
Câu 5. B
Câu 6. A
Câu 7. C
Câu 8. C
Câu 9. D
Câu 10. A
Câu 11. A
Câu 12. A
Câu 13. D
Câu 14. D

Câu 15. D
Câu 16. A

Câu 17. A
Câu 18. C
Câu 19. D
Câu 20. D
Câu 21. A
Câu 22. A
Câu 23. C
Câu 24. A

Câu 25. C

Câu 26. A
Câu 27. D

Câu 28. D

Câu 29. A
Câu 30. D
Câu 31. C
Câu 32. C

Câu 33. A

Câu 34. A

Câu 35. B

Câu 36. A

Câu 37. B

Câu 38. A

Câu 39. B

Câu 40. C

Câu 41. C

Câu 42. B

Câu 43. D

Câu 44. B

Câu 45. A

Câu 46. A


Câu 47. A

Câu 48. A

Câu 49. A

Câu 50. A

Giảng viên ra đề: (Chữ ký và Họ tên)	(Ngày ra đề)	Người phê duyệt: (Chữ ký và họ tên)	(Ngày duyệt đề)
--	--------------	---	-----------------

 TRƯỜNG ĐH BÁCH KHOA - ĐHQG-HCM <u>KHOA KH & KT MÁY TÍNH</u>	THI CUỐI KỲ		Học kỳ / Năm học		2	2023-2024
			Ngày thi		22-05-2024	
	Môn học	Kỹ thuật lập trình				
	Mã môn học	CO1027				
	Thời lượng	90 phút	Mã đề	2354		
Ghi chú: <ul style="list-style-type: none">- Đề thi có tổng cộng 60 câu.- Sinh viên được phép sử dụng một tờ A4 tài liệu tham khảo viết tay và máy tính cầm tay.- Các thư viện cơ bản đã được include đầy đủ.- Nộp lại đề sau khi kiểm tra/thi.						

Câu 1. [L.O.2.3] Hãy cho biết kết quả xuất ra màn hình của đoạn chương trình sau:

```
int x=1;
int *p=&x;
int &q=x;
cout << (q == *p);
```

- ☐ (A) Lỗi biên dịch
 ☐ (B) 0
 ☐ (C) Địa chỉ của x và giá trị của x
☐ (D) 1

Câu 2. [L.O.2.1] Độ phức tạp của hàm đệ quy sau là:

```
int recursiveFunction(int n) {
    if (n <= 1) {
        return 1;
    }
    return recursiveFunction(n - 1) + recursiveFunction(n - 2);
}
```

- ☐ (A) $O(\log n)$
 ☐ (B) $O(n!)$
 ☐ (C) $O(2^n)$
 ☐ (D) $O(n^2)$

Câu 3. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main(int argc, char const *argv[]) {
    string str;
    cin>>str;
    cout<<str;
    return 0;
}
```

- ☐ (A) Error (Lỗi)
 ☐ (B) str
☐ (C) Chuỗi nhập vào bởi người dùng
 ☐ (D) Garbage value (giá trị rác)

Câu 4. [L.O.1.1] Hãy cho biết output của mã giả sau:

```
Set matrix to [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Set finalMatrix to empty list

For i from 0 to 2:
    Set newRow to empty list
    For j from 0 to 2:
        Append matrix[2-j][i] to newRow
    End For
    Append newRow to finalMatrix
End For
```

- (A) [[1, 4, 7], [2, 5, 8], [3, 6, 9]] (B) [[9, 6, 3], [8, 5, 2], [7, 4, 1]]
(C) [[3, 6, 9], [2, 5, 8], [1, 4, 7]] (D) [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

Câu 5. [L.O.3.1] Cho đoạn code dưới đây:

```
class Node {
public:
    Node() { cout << "Normal Constructor called\n"; }
    Node(const Node &t) { cout << "Copy constructor called\n"; }
};

int main() {
    Node *t1, *t2; // Line 1
    t1 = new Node(); // Line 2
    t2 = new Node(*t1); // Line 3
    Node t3 = *t1; // Line 4
    return 0;
}
```

Chọn câu SAI:

- (A) Copy constructor được gọi ở dòng 4 (B) Normal constructor được gọi ở dòng 2
(C) Copy constructor được gọi ở dòng 3 (D) Normal constructor được gọi ở dòng 1

Câu 6. [L.O.2.3] Kết quả của đoạn code C bên dưới là gì

```
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *ptr = arr;
    printf("%d", *(ptr + 3));
    return 0;
}
```

- (A) Lỗi runtime (B) 2 (C) 4 (D) 5

Câu 7. [L.O.3.2] Thế nào được gọi là hiện tượng nạp chồng (overriding)?

- (A) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng các đối số nhưng khác kiểu với một hàm ở lớp cha
(B) Hiện tượng lớp con kế thừa định nghĩa một hàm hoàn toàn giống lớp cha
(C) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên, cùng kiểu với một hàm ở lớp cha nhưng khác đối số
(D) Hiện tượng lớp con kế thừa định nghĩa một hàm cùng tên nhưng khác kiểu với một hàm ở lớp cha

Câu 8. [L.O.2.4] Kết quả chương trình sau là gì?

```
#define foo(a, b) ((a) > (b) ? (a) : (b))
int main() {
    int x = 10, y = 20;
    cout << foo(x, y) << endl;
    return 0;
}
```

- (A) Lỗi runtime (B) Lỗi biên dịch (C) 20 (D) 10

Câu 9. [L.O.1.1] Chọn phát biểu đúng về mô tả giải thuật:

- (A) Có nhiều cách khác nhau để mô tả giải thuật (B) Cần phải dùng mã lập trình
(C) Cần phải dùng mã giả (pseudocode) (D) Cần phải dùng lưu đồ (flowchart)

Câu 10. [L.O.2.1] Cho đoạn mã chương trình sau của một hàm đệ qui và cách sử dụng hàm:

```
void recur_func(int n) {
    if (n<=1)
        return;
    cout << n << " -> ";
    if(n % 2)
        recur_func(3*n+1);
    else
        recur_func(n/2);
}
int main() {
    int n = 5;
    recur_func(n);
    return 0;
}
```

Kết quả khi chạy hàm *main()* là:

- (A) 5 -> 4 -> 2 -> 16 -> 8 -> (B) 5 -> 16 -> 8 -> 4 -> 2 ->
(C) 5 -> 2 -> 16 -> 2 -> 4 -> (D) Runtime error (Lỗi chạy chương trình)

Câu 11. [L.O.3.1] Khi xây dựng một lớp trong C++, nếu không xác định cụ thể, các thuộc tính và phương thức của lớp đó sẽ có tầm vực mặc định là:

- (A) package (B) protected (C) public (D) private

Câu 12. [L.O.2.1] Kết quả của *fun(2)* là gì?

```
int fun(int n) {
    if (n == 4)
        return n;
    else return 2*fun(n+1);
}
```

- (A) 8 (B) 4 (C) 32 (D) 16

Câu 13. [L.O.2.2] Trong C++, cho các string *s1*, *s2* và *s3* được khai báo như sau:

```
string s1 = "Hello_World";
string s2(s1, 3);
string s3(s2, 1, 4);
```

Giá trị của string *s3* là?

- (A) _Wor (B) o_Wo (C) ello (D) orld

Câu 14. [L.O.3.1] Một đối tượng trong lập trình hướng đối tượng bao gồm:

- (A) Thuộc tính và phương thức (B) Hằng số và biến
(C) Toán tử và hàm (D) Cấu trúc dữ liệu

Câu 15. [L.O.2.4] Kết quả khi chạy đoạn chương trình sau là:

```
int A[] = {2, 4, 1, 6, 8, 3, 1, 10};
int *p = A;
while (p < A + 6) {
    cout << --*p++ << *p++;
    p++;
}
```

- (A) 2367 (B) 1458 (C) 3158 (D) 4168

Câu 16. [L.O.3.1] Câu nào sau đây mô tả chính xác hàm ảo trong lập trình hướng đối tượng?

- (A) Một hàm có thể được gọi mà không cần một thể hiện đối tượng.
(B) Một hàm có thể được ghi đè bởi (các) lớp kế thừa để cung cấp cách triển khai cụ thể.
(C) Hàm được gọi tự động khi một đối tượng được tạo.
(D) Một hàm được khai báo bằng từ khóa virtual chỉ có thể được truy cập trong lớp nơi nó được khai báo.

Câu 17. [L.O.2.3] Cách truy xuất vào một thành phần của đối tượng con trở this?

- (A) this.x (B) this_x (C) this->x (D) (*this)->x

Câu 18. [L.O.2.2] Phát biểu nào sau đây là ĐÚNG về Struct trong C++

- (A) Struct không thể có phương thức
(B) Tầm vực truy xuất mặc định của các phần tử trong struct là private
(C) Có thể thay đổi tầm vực truy xuất của các thành phần trong struct
(D) Struct không hỗ trợ kế thừa

Câu 19. [L.O.2.2] Trong ngôn ngữ lập trình C++, để khai báo một mảng có kích thước cố định và lưu trữ các số nguyên, ta sử dụng cú pháp nào?

- (A) `int[] myArray = new int[10];` (B) `array<int> myArray(10);`
(C) `array<int, 10> myArray;` (D) `int myArray[10];`

Câu 20. [L.O.2.2] Kết quả in ra của đoạn chương trình sau, nếu chuỗi nhập vào ở lệnh cin là "I love BachKhoa":

```
string st;
cin >> st;
cout << st;
```

- (A) I (B) I love BachKhoa (C) BachKhoa (D) I love

Câu 21. [L.O.2.3] Trong đoạn code C++ dưới đây, giá trị nào sẽ được in ra sau khi chạy chương trình?

```
int main() {
    char str[] = "hcmut-cse";
    char *ptr = str;
    ptr += 5;
    cout << ptr;
    return 0;
}
```

- (A) cse (B) t-cse
(C) -cse (D) Chương trình bị lỗi biên dịch (Compilation Error)

Câu 22. [L.O.3.2] Trong lập trình hướng đối tượng (OOP), encapsulation có ý nghĩa gì?

- Ⓐ Quy tắc cho phép một đối tượng tự quản lý bộ nhớ
- Ⓑ Quy tắc cho phép một đối tượng che giấu thông tin bên trong và chỉ cho phép truy cập thông qua các phương thức
- Ⓒ Quy tắc cho phép một đối tượng kết nối với các đối tượng khác
- Ⓓ Quy tắc cho phép một đối tượng kế thừa các thuộc tính và phương thức từ một đối tượng khác

Câu 23. [L.O.2.1] Cho hiện thực hóa 1 hàm như sau:

```
void func(int* const list, int arraySize) {  
    for (int i=0; i<arraySize; i++)  
        list[i] = list[i] + 1;  
}  
  
int main() {  
    int numbers[5] = {5, 2, 7, 7, 1};  
    func(numbers, 5);  
    for (int i=0; i < 5; i++)  
        cout << numbers[i];  
    return 0;  
}
```

The value of array *numbers* when running the *main()* function is:

- Ⓐ Lỗi khi chạy
- Ⓑ {5, 2, 7, 7, 1}
- Ⓒ {6, 3, 8, 8, 2}
- Ⓓ Lỗi biên dịch

Câu 24. [L.O.2.2] Trong C++, ta khai báo một mảng số nguyên hai chiều như sau: `int arr[5][7];`

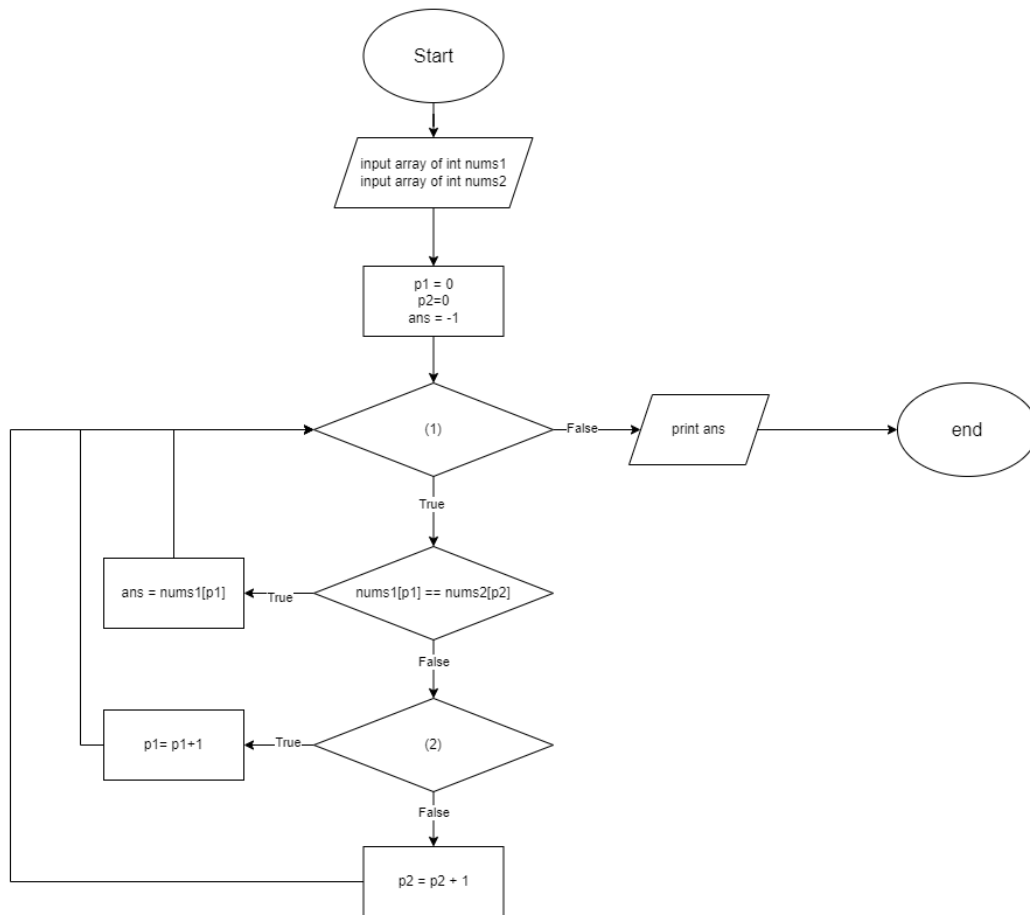
Kích thước của một giá trị số nguyên là 4 byte. Địa chỉ ô nhớ bắt đầu của arr là 0x70, địa chỉ vùng nhớ của arr[3][4] là bao nhiêu?

- Ⓐ 0x8c
- Ⓑ 0xa0
- Ⓒ 0xfc
- Ⓓ 0xd4

Câu 25. [L.O.2.3] Cách khai báo biến tham chiếu ref (reference variable) trong C++ là gì?

- Ⓐ int &ref;
- Ⓑ ref *int;
- Ⓒ int *ref;
- Ⓓ ref int;

Câu 26. [L.O.1.2] Để thực hiện thuật toán tìm phần tử nhỏ nhất *cùng xuất hiện* trong cả 2 mảng được sắp xếp theo thứ tự *tăng dần*, một thuật toán được thiết kế theo flowchart sau, hãy điền vào vị trí (1) và (2) sao cho chương trình sẽ in ra giá trị phần tử thỏa đề nếu có và ngược lại in -1 (Lưu ý, `nums1.size()` là lấy số phần tử có trong mảng `nums1`)



- (A) (1): `ans == -1 || (p1 < nums1.size() && p2 < nums2.size())` (2): `nums1[p1] > nums2[p2]`
 (B) (1): `ans == -1 && p1 < nums1.size() && p2 < nums2.size()` (2): `nums1[p1] < nums2[p2]`
 (C) (1): `ans == -1 && p1 < nums1.size() && p2 < nums2.size()` (2): `nums1[p1] > nums2[p2]`
 (D) (1): `ans == -1 || (p1 < nums1.size() && p2 < nums2.size())` (2): `nums1[p1] < nums2[p2]`

Câu 27. [L.O.1.1] Hãy cho biết output của mã giả sau:

```

Set numList to [4, 6, 0, 1, 8, 3, 9]
Set res to numList[0]

```

```

For i from 1 to length of numList - 1:
    If numList[i] > res:
        Set res to numList[i-1]
End For

```

Output res

- (A) Các đáp án khác đều sai
 (B) 8
 (C) 3
 (D) 9

Câu 28. [L.O.3.2] Class B sử dụng được các biến thành viên nào của class A:

```
class A {  
    int a, b;  
    protected:  
        double c, d;  
};  
class B : public A {  
};
```

- (A) Không truy cập được biến nào
(B) a, b
(C) a, b, c, d
(D) c, d

Câu 29. [L.O.2.3] Dòng nào trong đoạn code dưới đây là KHÔNG hợp lệ:

```
int a = 10;  
int b = 11;  
const int* p = &a; // Line 3  
*p = 100; // Line 4  
p = &b; // Line 5
```

- (A) Tất cả đều hợp lệ
(B) Dòng 3
(C) Dòng 4
(D) Dòng 5

Câu 30. [L.O.2.3] Mục đích của con trỏ trong lập trình là gì

- (A) Cấp phát bộ nhớ động
(B) Lưu trữ địa chỉ của biến
(C) Thực hiện các phép tính số học
(D) Lưu trữ giá trị của biến

Câu 31. [L.O.2.3] Câu nào mô tả đúng nhất về memory leak?

- (A) Chương trình không giải phóng được bộ nhớ không sử dụng, khiến bộ nhớ khả dụng dần cạn kiệt.
(B) Chương trình cố tình phân bổ nhiều bộ nhớ hơn mức cần thiết.
(C) Lỗi lập trình dẫn đến hỏng dữ liệu được lưu trữ trong bộ nhớ.
(D) Chương trình sử dụng bộ nhớ không hiệu quả, dẫn đến hiệu suất chậm hơn.

Câu 32. Biểu tượng hình thoi trong sơ đồ luồng (flowchart) biểu diễn cho:

- (A) Điểm bắt đầu hoặc điểm kết thúc của luồng đó
(B) Một điểm nơi mà có một câu hỏi được đặt và luồng được phân nhánh dựa vào câu trả lời
(C) Một quy trình hoặc một hành động cụ thể
(D) Một liên kết để kết nối các phần khác nhau của sơ đồ luồng

Câu 33. [L.O.3.2] Cho lớp A là cha của lớp B, và biến a có kiểu A, biến b có kiểu B.

Phát biểu gán 1: a = b;

Phát biểu gán 2: b = a;

- (A) Cả hai phát biểu gán đều đúng
(B) Phát biểu 1 đúng, 2 sai
(C) Cả hai phát biểu gán đều đúng
(D) Phát biểu 1 sai, 2 đúng

Cho chương trình như sau; giả sử rằng các thư viện cần thiết đã được **include**. Dùng đoạn mã này cho các câu hỏi từ 34 - 35.

```
/*Mark-1*/ A {  
    A(int value=9): value_(value){ cout << "A, "; }  
    ~A(){ cout << "~A, "; }  
/*Mark-2*/  
    int value_;  
};  
int main() {  
    cout << "[";
```

```

    {A a; {A a; cout << "x, ";} cout << a.value_ << ", "; }
    cout << "]" ; return 0;
}

```

Câu 34. Thay thế /*Mark-1*/ bởi **struct** và /*Mark-2*/ bởi **public**:. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- ☐ (A) Kết quả in ra là: [A, A, x]
☐ (B) Có lỗi trong **main**, vì có hai biến cùng tên là **a**.
☐ (C) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private.
☐ (D) Tất cả các lựa chọn khác đều sai.

Câu 35. Thay thế /*Mark-1*/ bởi **class** và /*Mark-2*/ bởi **public**:. Khi biên dịch và chạy chương trình; lựa chọn nào sau đây là đúng?

- ☐ (A) Kết quả in ra là: [A, A, x, ~A, 9, ~A,]
☐ (B) Có lỗi trong **main**, vì có hai biến cùng tên là **a**.
☐ (C) Có lỗi trong **main**, vì phương thức khởi tạo và hủy có quyền private.
☐ (D) Tất cả các lựa chọn khác đều sai.

Câu 36. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```

int array1[] = {1, 2, 3, 4, 5};
int array2[] = {6, 7, 8, 9, 10};
int temp, result = 0;
int main() {
    for (temp = 0; temp < 5; temp++) {
        result += array1[temp];
    }
    for (; temp < 4; temp++) {
        result += array2[temp];
    }
    cout << result;
    return 0;
}

```

- ☐ (A) 45
☐ (B) 55
☐ (C) 15
☐ (D) 35

Thông tin sau áp dụng cho câu hỏi 37:

```
int i = 0;
```

```

class A {
public:
    A() {
        i = 3;
    }
    ~A() {
        i=10;
    }
};

```

```

int foo() {
    int &p = i;
    A ob;
    return p;
}

```

Câu 37. [L.O.3.2] Lời gọi hàm *foo()* sẽ trả về giá trị nào?

- (A) Phương án khác (B) 0 (C) 10 (D) 3

Câu 38. [L.O.2.2] Kết quả khi chạy chương trình sau là gì?

```
int main() {
    int a = 1, b = 10, c = 3;
    int *arr[3] = {&a, &b, &c};
    cout << *arr[*arr[0] * 2] - c;
    return 0;
}
```

- (A) Lỗi biên dịch (B) 3 (C) Giá trị rác (D) 0

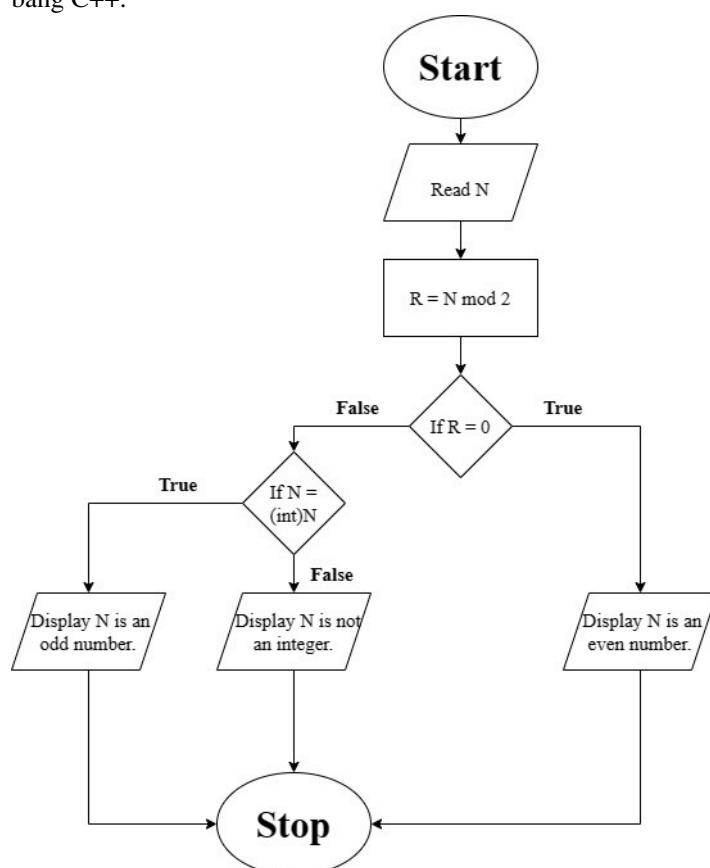
Câu 39. [L.O.3.1] Chọn phát biểu SAI trong các phát biểu sau:

- (A) Hàm khởi tạo có thể là một hàm ảo (virtual function) trong abstract class (B) Một class chỉ có thể có duy nhất một hàm hủy (destructor)
(C) Hàm khởi tạo có tham số truyền vào, nhưng hàm hủy thì không (D) Một class có thể có nhiều hàm khởi tạo (constructor)

Câu 40. [L.O.2.3] Trong lập trình C, con trỏ **void*** có thể đại diện cho cái gì?

- (A) Con trỏ đến địa chỉ của kiểu dữ liệu bất kỳ (B) Con trỏ đến địa chỉ của một biến kiểu void
(C) Con trỏ đến một biến hằng (D) Con trỏ đến một hàm có kiểu trả về là void

Câu 41. [L.O.1.2] Trường hợp nào không thể xảy ra với lưu đồ được cho bên dưới. Giả sử lưu đồ sẽ được hiện thực bằng C++.



- (A) Đầu vào: N = 3.5, Đầu ra: N is not an integer (B) Đầu vào: N = 1, Đầu ra: N is an odd number
(C) Đầu vào: N = 4, Đầu ra: N is an even number (D) Tất cả các câu trên đều có thể xảy ra.

Câu 42. [L.O.3.1] Điều nào sau đây mô tả đúng nhất các lớp trừu tượng?

- (A) Nếu một lớp chỉ có tất cả các hàm ảo thuần túy thì đó là lớp trừu tượng
- (B) Nếu một lớp có nhiều hơn một hàm ảo thì đó là lớp trừu tượng
- (C) Nếu một lớp chỉ có một hàm ảo thuần túy thì đó là lớp trừu tượng
- (D) Nếu một lớp có ít nhất một hàm ảo thuần túy thì đó là lớp trừu tượng

Thông tin sau áp dụng cho câu hỏi 43:

```
class Triangle {
public:
    static int numOfTriangle;
private:
    int a;
    int b;
    int c;
public:
    Triangle(int a, int b, int c) : a(a), b(b), c(c) {
        numOfTriangle++;
    }
    ~Triangle() {
        numOfTriangle--;
        cout << 3;
    }
};

int Triangle::numOfTriangle = 0;
int main() {
    Triangle aList[5];
    cout << (*aList).numOfTriangle;
    return 0;
}
```

Câu 43. [L.O.3.1] Cho biết kết quả in ra màn hình của chương trình:

- (A) Lỗi biên dịch
- (B) 5
- (C) 533333
- (D) 53

Câu 44. [L.O.1.2] Ký hiệu nào dưới đây trong lưu đồ (flowchart) được dùng để biểu thị output được trả về?

- (A) Hình thoi
- (B) Hình bình hành
- (C) Hình tròn/ellipse
- (D) Hình chữ nhật

Các thông tin sau sử dụng cho câu 45. Một nhân vật có một túi đồ để chứa các vật phẩm có thể được thu thập trong khi di chuyển trên bản đồ. Class Character biểu diễn một nhân vật. Một Character có 2 thuộc tính private là hp và exp. Class Character cũng có các cặp hàm get, set để lấy và gán giá trị của thuộc tính tương ứng. Class BaseBag biểu diễn một túi đồ có mã nguồn được cho bên dưới. Abstract class BaseItem biểu diễn một vật phẩm. Túi đồ được hiện thực bằng danh sách liên kết, mỗi phần tử của danh sách là một Node, head và tail lần lượt là Node đầu tiên và Node cuối cùng của danh sách.

```
class Character {
private:
    int hp, exp;
public:
    int getHp() const;
```

```

    void setHp(int hp);
    int getExp() const;
    void setExp(int exp);
}

class BaseItem {
// Definition of class BaseItem
};

class Node {
friend class BaseBag;
private:
    BaseItem * item;
    Node * next;
public:
    Node(BaseItem * item, Node * next = NULL) {
        this->item = item;
        this->next = next;
    }
    ~Node() {}
};

class BaseBag {
private:
    Node * head, * tail;
    int capacity;
    int count;
protected:
    virtual void removeFirst() {
        if (count == 1) {
            delete head;
            head = NULL;
            tail = NULL;
        }
        else {
            /* r1 */
            /* r2 */
            /* r3 */
        }
        --count;
    }
public:
    BaseBag(int capacity) {
        this->capacity = capacity;
        this->count = 0;
        this->head = this->tail = NULL;
    }
};

```


Câu 45. [L.O.2.3] Phương thức `removeFirst(Character * ch)` của `BaseBag` thực hiện xóa vật phẩm đầu tiên của túi đồ nếu túi đồ có ít nhất 1 vật phẩm (giả sử khi được gọi, phương thức luôn có ít nhất 1 vật phẩm). Chọn đoạn mã hợp lệ để điền vào các ghi chú (comment) `/* r1 */`, `/* r2 */`, `/* r3 */`:

- (A) `/* r1 */ Node * temp = head; /* r2 */ delete head; /* r3 */ head = temp->next;`
- (B) `/* r1 */ Node * temp = this->head; /* r2 */ head = head->next; /* r3 */ delete temp;`
- (C) `/* r1 */ head = head->next; /* r2 */ delete head; /* r3 */` (không cần điền mã)
- (D) `/* r1 */ delete head; /* r2 */ head = head->next; /* r3 */` (không cần điền mã)

Các thông tin sau sử dụng cho câu 46 đến 48:

Cho struct `PasswordInfo` mô tả thông tin về một mật khẩu trong mảng `arr_pwds[]` cụ thể như sau:

```
struct PasswordInfo {
    char pwd[50];
    int freq;
    int first_index;
};
```

Trong đó:

- `char pwd[50]`: Chuỗi password

- `int freq`: tần suất xuất hiện của password trong mảng `arr_pwds[]`

- `first_index`: Vị trí index xuất hiện đầu tiên của password

Hàm `void buildArrayUniquePassword>PasswordInfo arr_unique_pwds[], int & num_unique_pwds, const char * arr_pwds[], int num_pwds)` để tạo ra mảng các `PasswordInfo` không trùng nhau từ mảng các password trong `arr_pwds`.

Tham số đầu vào gồm:

- `PasswordInfo arr_unique_pwds[]`: Mảng các `PasswordInfo` không trùng nhau

- `int & num_unique_pwds`: Giá trị tham chiếu số lượng password không trùng nhau

- `arr_pwds` và `num_pwds` lần lượt là mảng các password và số lượng phần tử của mảng

Nội dung hàm như sau:

```
void buildArrayUniquePassword>PasswordInfo arr_unique_pwds[], int &
num_unique_pwds, const char * arr_pwds[], int num_pwds) {
    bool first_meet;
    num_unique_pwds = 0;
    //First loop
    for (int i_f = 0; i_f < num_pwds; ++i_f) { // index_full
        first_meet = true;
        //Second loop
        for (int iu = 0; iu < num_unique_pwds; ++iu) { // index_unique
            if (strcmp(arr_pwds[i_f], arr_unique_pwds[iu].pwd) == 0) {
                ++arr_unique_pwds[iu].freq;
                first_meet = false;
                break;
            }
        }
        //missing_stmt_a
        {
            strcpy(arr_unique_pwds[num_unique_pwds].pwd, arr_pwds[i_f]);
            arr_unique_pwds[num_unique_pwds].first_index = i_f;
            //missing_stmt_b
            ++num_unique_pwds;
        }
    }
}
```

```
}
```

Câu 46. [L.O.2.2] Nếu thiếu câu lệnh `num_unique_pwds = 0` bên trong như hàm trên điều gì sẽ xảy ra?

- (A) Số lần lặp nhiều hơn nhưng kết quả vẫn chính xác
- (B) Chương trình có thể sẽ gây ra lỗi
- (C) Giá trị `freq` của `PasswordInfo` tương ứng sẽ luôn luôn không được cập nhật
- (D) Chương trình sẽ chỉ lặp vòng lặp `//Second loop` qua đúng 1 lần tại phần tử đầu tiên của mảng `arr_unique`

Câu 47. [L.O.2.2] Nội dung tại `//missing_stmt_a` có thể là:

- (A) `if (!first_meet || !num_unique_pwds)`
- (B) `if (first_meet)`
- (C) `if (first_meet || num_unique_pwds)`
- (D) `if(!first_meet)`

Câu 48. [L.O.2.2] Với nội dung đã chọn cho `//misng_stmt_a`, vậy nội dung câu lệnh có thể ở `//missing_stmt_b` là gì

- (A) Đáp án A và B đều đúng
- (B) `arr_unique_pwds[num_unique_pwds].freq = 1;`
- (C) `arr_unique_pwds[num_unique_pwds].freq = -1;`
- (D) `arr_unique_pwds[num_unique_pwds].freq ++;`

Các thông tin sau sử dụng cho câu 49 đến 50:

Một bản đồ được biểu diễn bằng mảng 2 chiều gồm `n_rows` (một biến kiểu `int`) là số hàng và `n_cols` (một biến kiểu `int`) là số cột. Mỗi phần tử của mảng là một đối tượng thuộc class `MapElement`. Trên bản đồ có 2 loại phần tử là đường đi (biểu diễn bằng class `Path`) và tường (biểu diễn bằng class `Wall`). `Path` có thể cho phép một nhân vật di chuyển được trong khi tường thì không cho phép nhân vật di chuyển.

```
/* a */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    map[r] = new MapElement*[n_cols];
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Path();
    }
}
```

```
/* b */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    map[r] = new MapElement*[n_cols];
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}
```

```
/* c */
MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Path();
    }
}
```

```
/* d */
```

```

MapElement *** map = new MapElement**[n_rows];
for (int r = 0; r < n_rows; ++r) {
    for (int c = 0; c < n_cols; ++c) {
        map[r][c] = new Wall();
    }
}

```

Câu 49. [L.O.2.3] Biền map được sử dụng để biểu diễn bản đồ. Giả sử mảng 2 chiều cho bản đồ được cấp phát động. Cho yêu cầu cần khởi tạo một bản đồ mà tất cả các phần tử đều là Path. Biết rằng Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu này:

- ☐ (A) Đoạn code bên dưới comment /* d */
 ☐ (B) Đoạn code bên dưới comment /* a */
 ☐ (C) Đoạn code bên dưới comment /* c */
 ☐ (D) Đoạn code bên dưới comment /* b */

Câu 50. [L.O.2.3] Biền map được sử dụng để biểu diễn bản đồ. Biết rằng: hiện tại bản đồ gồm tất cả các phần tử đều là Path; Wall và Path đều có hàm khởi tạo (Constructor) không tham số. Cho yêu cầu: tạo ra một bản đồ gồm tất cả phần tử ở hàng đầu tiên đều là Wall. Chọn đáp án là đoạn code hợp lệ để thực hiện yêu cầu trên và không tạo ra rò rỉ bộ nhớ (memory leak):

```

/* a */
for (int i = 0; i < n_cols; ++i) {
    map[0][i] = new Wall();
}

/* b */
for (int i = 0; i < n_cols; ++i) {
    delete map[0][i];
    map[0][i] = new Wall();
}

/* c */
for (int i = 0; i < n_rows; ++i) {
    delete map[i][0];
    map[i][0] = new Wall();
}

/* d */
for (int i = 0; i < n_cols; ++i) {
    delete (*map[n_rows-1][i]);
    map[n_rows-1][i] = new Wall();
}

```

- ☐ (A) Đoạn code bên dưới comment /* d */
 ☐ (B) Đoạn code bên dưới comment /* b */
 ☐ (C) Đoạn code bên dưới comment /* c */
 ☐ (D) Đoạn code bên dưới comment /* a */

===== Kết thúc đề thi =====

Câu 1. D

Câu 2. C

Câu 3. C

Câu 4. B

Câu 5. D

Câu 6. C

Câu 7. B

Câu 8. C

Câu 9. A

Câu 10. B

Câu 11. D

Câu 12. D

Câu 13. B

Câu 14. A

Câu 15. B

Câu 16. B

Câu 17. C

Câu 18. C

Câu 19. D

Câu 20. A

Câu 21. C

Câu 22. B

Câu 23. C

Câu 24. D

Câu 25. A

Câu 26. B

Câu 27. C

Câu 28. D

Câu 29. C

Câu 30. B

Câu 31. A

Câu 32. B

Câu 33. B

Câu 34. D

Câu 35. C

Câu 36. C

Câu 37. D

Câu 38. D

Câu 39. A

Câu 40. A

Câu 41. D

Câu 42. D

Câu 43. A

Câu 44. B

Câu 45. B

Câu 46. B

Câu 47. B

Câu 48. B

Câu 49. B

Câu 50. B