



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM  
**ĐỀ THI GIỮA KỲ**  
**Học kỳ 2 – Năm học 2020-2021**

**MÃ LƯU TRỮ**  
(do phòng KT-ĐBCL ghi)

Tên học phần: Kỹ thuật lập trình Mã HP: CSC10002  
Thời gian làm bài: 60 phút Ngày thi: 27/04/2021  
Ghi chú: Sinh viên [☒ được phép / ☐ không được phép] sử dụng tài liệu khi làm bài.

Họ tên sinh viên: ..... MSSV: ..... STT: .....

**Câu 1:** Cho đoạn chương trình sau

```
int foo = 0;  
int* ptr = &foo;  
int*& ref1 = ptr;  
int& ref2 = foo;
```

Câu lệnh nào sau đây sẽ thay đổi giá trị của biến `foo` thành 1? Giải thích ngắn gọn

A. `ptr++`; B. `foo++`; C. `(*foo)++`; D. `(*ptr)++`; E. `ref2++`; F. `(*ref1)++`;

**Câu 2:** Cho đoạn chương trình sau:

<pre>const int LIMIT = 50; // Dòng 1 int AddEm(int x, int y); // 2 void main() { // 3     int x = 42, y = 35 // 4     int Sum; // 6     Sum = AddEm(x, y); // 7     Sum++; // 8 } // 9</pre>	<pre>int AddEm(int x, int y) { // 10     static int Total; // 11     Total = x + y; // 12     if (Total &gt; LIMIT) // 13         Total = 0; // 14     return (Total); // 15 } // 16</pre>
--	--

- a. Cho biết Tầm vực (Scope) của biến `Sum` được khai báo ở dòng 6 là từ dòng nào tới dòng nào? Giải thích ngắn gọn
- b. Cho biết Thời gian sống (Lifetime) của biến `Total` được khai báo ở dòng 11 là bao lâu? Giải thích ngắn gọn

**Câu 3:** Cho biết đoạn chương trình sau xuất ra kết quả là gì?

<pre>void func(int* a, int&amp; b) {     int w;     *a = *a + *a;     w = *a;     *a = b;     b = w; }</pre>	<pre>void main() {     int x = 2, y = 5;     int* px = &amp;x;     func(px, y);     cout &lt;&lt; x &lt;&lt; ", " &lt;&lt; y &lt;&lt; endl; }</pre>
--	---



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM  
**ĐỀ THI GIỮA KỲ**  
**Học kỳ 2 – Năm học 2020-2021**

**MÃ LƯU TRỮ**  
(do phòng KT-ĐBCL ghi)

**Câu 4:** Cho đoạn chương trình dùng để tính trung bình cộng của mảng số nguyên

```
double Average(const int* const Data, int Sz) {           // Dòng 1
    if (Data == NULL || Sz <= 0) return 0.0;             // 2
    int Sum = 0;                                         // 3
    const int *Current = Data;                          // 4
    for (int Pos = 0; Pos < Sz; Pos++) {                 // 5
        Sum = Sum + _____;                         // 6
        Current++;                                       // 7
    }                                                    // 8
    return (double(Sum) / Sz);                          // 9
}
```

- Giải thích sự khác nhau về tác dụng của 2 từ khóa **const** được khai báo ở dòng 1
- Có thể bỏ đi từ khóa **const** ở dòng 4 hay không? Vì sao?
- Sử dụng biến **Current** ở dòng 4 hoàn tất câu lệnh dòng 6 để đoạn chương trình chạy đúng (chỉ cần ghi lại dòng code 6, không cần ghi lại cả đoạn)

**Câu 5:** Cho biết hàm func dưới đây bị lỗi gì? Hãy sửa lại cho đúng

```
int* func(int* a, int c) {
    int* ptr = &c;
    *ptr = *a;
    return ptr;
}
```

**Câu 6:** Cho biết đoạn chương trình sau bị lỗi gì? Hãy sửa lại cho đúng

<pre>struct MyArr {     int* p; };</pre>	<pre>void main() {     MyArr* a = new MyArr;     a-&gt;p = new int[100];     delete a; }</pre>
--	--

**Câu 7:** Cho đoạn chương trình sau

```
int a[3] = { 5, 10, 15 };
int* p = &a[0], *q = a + 1;
```

Các phần tử **a[0]**, **a[1]**, **a[2]**, và các giá trị **\*p**, **\*q** bằng bao nhiêu nếu một trong các dòng lệnh sau được thêm vào cuối?

- A. **\*p++=100;**                      B. **q[-1]++;**                      C. **\*q++ = \*p++;**



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM  
**ĐỀ THI GIỮA KỲ**  
**Học kỳ 2 – Năm học 2020-2021**

**MÃ LƯU TRỮ**  
(do phòng KT-ĐBCL ghi)

**Câu 8:** Dùng ngôn ngữ C/C++, viết hàm đọc mảng động số nguyên **A** có  $n$  phần tử và mảng động số nguyên **B** có  $m$  phần tử từ hai file nhị phân có chung cấu trúc, sau đó nối mảng **B** vào cuối mảng **A**, tạo thành mảng động **C** và ghi mảng **C** lên file nhị phân có cùng cấu trúc với hai file nhị phân lưu mảng **A** và mảng **B**. Cấu trúc file nhị phân lưu mảng động như sau

- 4 byte đầu tiên lưu số phần tử của mảng
- Các byte còn lại lần lượt lưu các phần tử trong mảng, mỗi phần tử chiếm 4 byte

Prototype hàm: `int Append(char* fileA, char* fileB, char* fileC);`

Trong đó: **fileA**, **fileB** là đường dẫn đến hai file nhị phân lưu mảng **A** và mảng **B**, **fileC** là đường dẫn của file nhị phân lưu mảng **C**. Hàm trả về 1 nếu thành công và trả về 0 nếu bị lỗi khi xử lý file.

Ví dụ:

- Mảng **A** gồm 3 phần tử 2, 3, 4 lưu trên **fileA** như sau

Số phần tử (4 byte)	A[0] (4 byte)	A[1] (4 byte)	A[2] (4 byte)
3	2	3	4

- Mảng **B** gồm 2 phần tử 12, 13 lưu trên **fileB** như sau

Số phần tử (4 byte)	B[0] (4 byte)	B[1] (4 byte)
2	12	13

- Mảng **C** kết quả sẽ có 5 phần tử 2, 3, 4, 12, 13 lưu trên **fileC** như sau

Số phần tử (4 byte)	C[0] (4 byte)	C[1] (4 byte)	C[2] (4 byte)	C[3] (4 byte)	C[4] (4 byte)
5	2	3	4	12	13