

Tên học phần: Kỹ thuật lập trình (Lớp 20CTT1) Mã HP: CSC10002  
Thời gian làm bài: 90 phút Ngày thi: 25/10/2021  
Ghi chú: Sinh viên [ ☒ được phép / ☐ không được phép ] sử dụng tài liệu **GIẤY** khi làm bài.

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

Lưu ý: - Hình vẽ/mã nguồn dùng C/C++ rõ ràng, sạch sẽ, đúng cú pháp

- Ngoài những hàm chính, có thể viết thêm các hàm phụ trợ nếu cần

**Câu 1 (3 điểm):** Xét đoạn mã sau (~25 phút)

```
1 void main(){
2     unsigned short a = -60234;
3     unsigned char* p = (unsigned char*)&a;
4     cout << short(*p) << endl;
5     cout << short(*(p + 1)) << endl;
6 }
```

Ứng với từng dòng mã 2, 3, 4, 5, hãy vẽ hình minh họa trong bộ nhớ (chính xác tới từng bit cho giá trị thông thường lẫn giá trị địa chỉ) cho các biến ‘a’ và ‘p’ cũng như các ép kiểu tương ứng của chúng. Lưu ý: không cần quan tâm tới biến ‘cout’ và toán tử ‘endl’

**Câu 2 (3 điểm):** Xét đoạn mã sau (~25 phút)

```
1 void swap(int& a, int& b){
2     int temp = a;
3     a = b;
4     b = temp;
5 }
6 void dummySort(int a[], int n){
7     for(int i = 0; i < n - 1; i++)
8         for(int j = i + 1; j < n; j++)
9             if(a[i] > a[j]) swap(a[i], a[j]);
10 }
```

Đoạn mã trên chỉ sắp xếp tăng dần, khi muốn sắp xếp theo tiêu chí khác ta cần viết lại hàm sắp xếp và thay đổi tại câu lệnh **if**. Ngoài ra, nếu muốn sắp xếp cho kiểu dữ liệu khác ta cũng cần lặp lại toàn bộ quá trình trên gây lãng phí mã.

Hãy dùng kỹ thuật **template** và **con trỏ hàm** để tổng quát hóa đoạn mã cho hàm “swap” và hàm “dummySort” sao cho có thể sắp xếp theo tiêu chí bất kỳ cho mọi kiểu dữ liệu (có thể phải xây dựng thêm các hàm phụ). Sau khi xây dựng xong đoạn mã tổng quát

cho hai hàm trên, hãy viết hàm “main” minh họa cách sử dụng để sắp xếp tăng và giảm cho mảng kiểu số nguyên.

**Câu 3 (3 điểm):** Trên một danh sách liên kết đơn chứa các số nguyên, hãy viết các hàm theo yêu cầu sau (~25 phút)

- Yêu cầu 1: **một hàm đệ quy kiểm tra** danh sách có chứa giá trị giảm đều hay không?

- Yêu cầu 2: **nâng cấp hàm trong “yêu cầu 1” thành một hàm kiểm tra đệ quy** sao cho có thể kiểm tra với điều kiện tổng quát, ví dụ tăng đều hay trị tuyệt đối giảm...

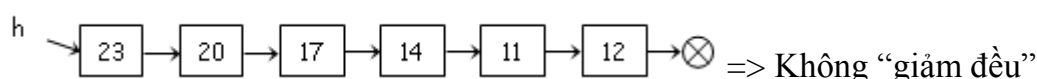
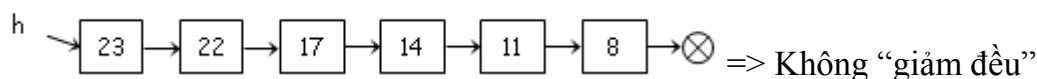
Chú ý:

+ Đoạn code sinh viên xây dựng **BẮT BUỘC** đều dựa vào cấu trúc sau để phát triển

```
struct Node{
    int data;
    Node* next;
};
```

+ Cả 2 yêu cầu đều phải thực hiện đệ quy, riêng “yêu cầu 2” kết hợp thêm kỹ thuật con trỏ hàm (**KHÔNG CẦN** tổng quát kiểu dữ liệu cho ‘data’)

+ Ví dụ:



**Câu 4 (1 điểm):** Viết hàm đệ quy tính toán theo biểu thức như bên dưới (~15 phút)

$$\begin{cases} T_0(x) = 1; T_1(x) = x \bmod p \\ T_n(x) = 2 \times x \times T_{n-1}(x) \bmod p - T_{n-2}(x) \bmod p \end{cases}$$

Trong đó ‘p’ là số nguyên tố,  $x \leq p$  và  $x, n \in \mathbb{N}$