TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM

ĐỀ THI KẾT THÚC HỌC PHẦN
Học kỳ 2 – Năm học 2020-2021

MÃ LƯU TRỮ (do phòng KT-ĐBCL ghi)
CK20212_CSC10002

Tên học phần:	**Kỹ thuật lập trình (Lớp 20CTT2)**	**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 khi làm bài.*

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

Lưu ý:

- Sinh viên sử dụng ngôn ngữ C/C++ để viết chương trình.
- Sinh viên phải dùng tên hàm và mẫu hàm được ghi trong đề. Đối với các hàm không có mẫu hàm, sinh viên phải tự thiết kế các tham số và giá trị trả về cho hàm.
- Sinh viên được phép tự thiết kế thêm các hàm nằm ngoài yêu cầu để phục vụ bài làm, nhưng phải ghi rõ ý nghĩa của các tham số và giá trị trả về của hàm.

Câu 1 (1 *điểm*)**.**

Cho đoạn mã nguồn sau

```
void main() {                                          //Dòng 1
      double *firstPtr = new double;                   //Dòng 2
      double *nextPtr = new double;                    //Dòng 3
      *firstPtr = 62;                                  //Dòng 4
      nextPtr = firstPtr;                              //Dòng 5
      delete firstPtr;                                 //Dòng 6
      delete nextPtr;                                  //Dòng 7
      firstPtr = new double;                           //Dòng 8
      *firstPtr = 28;                                  //Dòng 9
      cout << *firstPtr << " " << *nextPtr << endl;    //Dòng 10
}                                                      //Dòng 11
```

a. Chương trình chạy bởi đoạn mã nguồn trên xảy ra lỗi ở những dòng nào? Giải thích.

b. Viết lại đoạn mã nguồn trên và sửa các lỗi đã nêu.

Câu 2 (1 *điểm*)**.**

Viết hàm **`int countNumChar(char* str)`** đếm số lượng ký tự số (các ký tư ‘0’, ‘1’, ‘2’,…, ‘9’) có trong chuỗi bằng hai phương pháp:

a. Không dùng đệ quy

b. Có dùng đệ quy.

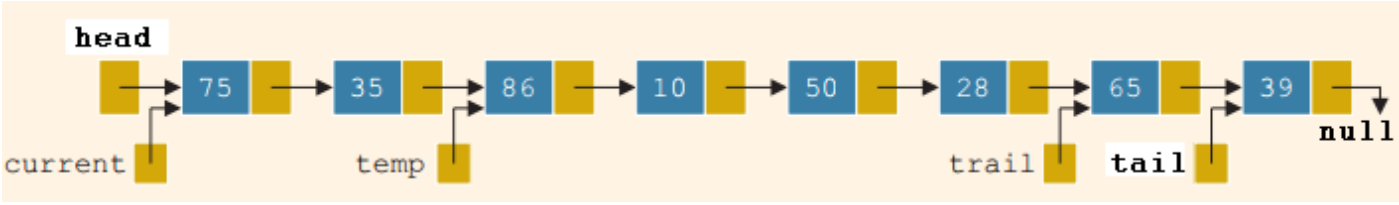
Ví dụ: chuỗi str = “**123** abd**5**cd efgh**78**” có số lượng ký tự số là 6

Câu 3 (4 *điểm*).

Cho cấu trúc **Node** biểu diễn một phần tử trong danh sách liên kết đơn gồm hai thành phần: trường dữ liệu (**data**) và con trỏ tới phần tử kế tiếp (**pNext**) được khai báo như sau

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

Một danh sách liên kết đơn được tạo ra như hình vẽ bên dưới trong đó **head, current, temp, trail**, và **tail** lần lượt là các con trỏ đến các node như trong hình



a. Dựa vào hình vẽ trên, cho biết kết quả in ra màn hình của đoạn chương trình sau

```
current = current->pNext;
cout << current->data;
cout << temp->pNext->pNext->data;
```

b. Dựa vào hình vẽ trên, cho biết điều gì xảy ra nếu đoạn chương trình sau được thực thi

```
trail->pNext = NULL;
delete tail;
```

c. Viết hàm **getMinNode** tìm node có trường data nhỏ nhất và hàm **getMaxNode** tìm node có trường data lớn nhất trong danh sách liên kết đơn

Mẫu hàm **Node* getMinNode(Node* head, Node* tail);**

Node* getMaxNode(Node* head, Node* tail);

d. Viết hàm **countNodeBetweenMinMax** đếm số lượng node nằm giữa node có trường data lớn nhất và node có trường data nhỏ nhất

Ví dụ: danh sách đầu vào

34 **456** 343 435 **4** 54

Kết quả trả về: 2

Câu 4 (4 *điểm*)**.**

Trường đại học A cần xây dựng phần mềm quản lý sinh viên. Thông tin của một sinh viên bao gồm:

- Mã số sinh viên: là chuỗi gồm 8 ký tự số (ví dụ: 20200001, 20200010,…)
- Họ và tên sinh viên: là chuỗi tối đa 100 ký tự
- Điểm trung bình tích lũy

Danh sách sinh viên lưu trong tập tin nhị phân **`STUDENTS.bin`** đã được **sắp xếp tăng dần theo mã số sinh viên**. Hai chuỗi mã số sinh viên được so sánh theo nguyên tắc so sánh lần lượt giá trị từng ký tự số ở cùng vị trí tương ứng từ trái sang phải:

- Nếu hai ký tự số cùng vị trí tương ứng của hai chuỗi có giá trị bằng nhau thì chuyển sang so sánh hai ký tự số kế tiếp cho đến khi xuất hiện ký tự số ở một chuỗi có giá trị lớn hơn hoặc nhỏ hơn ký tự số cùng vị trí ở chuỗi còn lại. Khi đó, ký tự số nào có giá trị lớn hơn thì chuỗi đó lớn hơn và ngược lại. Nếu hai chuỗi giống nhau hoàn toàn xem như bằng nhau.
- Ví dụ: “20**1**90001” < “20**2**00000”, “201**8**1234” > “201**7**0001”, “20175423” = “20175423”

Phần mềm có tính năng **thêm thông tin của một sinh viên mới** vào danh sách sinh viên trên tập tin **`STUDENTS.bin`** mà vẫn **giữ thứ tự tăng dần** theo mã số sinh viên của danh sách.

a. Viết hàm so sánh hai chuỗi mã số sinh viên.

Mẫu hàm: **`int compareStudentID(char* strID1, char* strID2);`**

Hàm trả về 0 nếu strID1 = strID2, trả về -1 nếu strID1 < strID2, và trả về 1 nếu strID1 > strID2.

b. Đề xuất định dạng phù hợp cho tập tin nhị phân **`STUDENTS.bin`**

c. So sánh hai cấu trúc dữ liệu **mảng động** và **danh sách liên kết đơn** dùng để lưu danh sách sinh viên trên bộ nhớ (RAM) theo các tiêu chí:

- Dung lượng lưu trữ trên bộ nhớ (RAM) khi thực hiện tính năng thêm sinh viên vẫn giữ thứ tự cho danh sách.
- Chi phí khi thực hiện tính năng thêm sinh viên vẫn giữ thứ tự cho danh sách.
- Số lần truy cập vào tập tin **`STUDENTS.bin`** khi thực hiện tính năng thêm sinh viên vẫn giữ thứ tự cho danh sách.

d. Sử dụng một trong hai cấu trúc: **mảng động** hoặc **danh sách liên kết đơn**, viết hàm **`addStudent`** thêm một sinh viên mới vào danh sách sinh viên trên tập tin **`STUDENTS.bin`** vẫn giữ thứ tự tăng dần theo mã số sinh viên của danh sách.