ĐỀ THI CUỐI KỲ - CA A

Thời gian làm bài: 60 phút

Quy định chung

- Đây là bài thi tại lớp. Sinh viên KHÔNG được sử dụng tài liệu, máy tính cá nhân, điện thoại di động và tham khảo internet, trừ khi có sự đồng ý của Giảng viên.
- Sinh viên không được trao đổi dưới bất kì hình thức nào.
- Sinh viên làm và nộp tất cả các bài trong cùng 1 file <**MSSV**>**.cpp**.
- Sinh viên cần chú thích bài làm rõ ràng.
- Sinh viên không cần kiểm tra tính hợp lệ của input.
- Các bài làm không biên dịch được, hoặc lỗi runtime đều sẽ bị **0 điểm**.
- Vi phạm 1 trong các quy định trên, sẽ bị **0 điểm** bài thi cuối kì.

(Sinh viên sang trang kế tiếp để xem đề bài.)

Nội dung

Lưu ý: Con trỏ rỗng được setup mặc định = NULL (không phải nullptr).

Câu 0

Cho file có tên `filename`, thông tin từ file này sẽ là dữ liệu cho các câu **1**, **2** và **3** phía dưới, dữ liệu của mỗi câu cách nhau 1 dòng trắng (xem file `data.txt` để biết thêm chi tiết). Hãy đọc dữ liệu từ file và lưu vào các biến (tham chiếu) của hàm `readfile`.

Lưu ý: Sinh viên cần đọc hết đề để biết từng dạng dữ liệu tương ứng trong file cần được trả về. Phần đọc file của mỗi câu sẽ có điểm riêng, đọc được dữ liệu của câu nào sẽ có một phần điểm tương ứng của câu đó.
Prototype:

```
void readfile(string filename, int &number, int** &matrixA, int &m, int &n, int** &matrixB,
                                            int &p, int &q, List& gene, List& toxic_gene)
```

Câu 1 *(2 điểm)*

Cho số tự nhiên n. Viết hàm phân tích n thành tích của các thừa số nguyên tố (tăng dần) và in kết quả ra màn hình (xem ví dụ).

Yêu cầu: bắt buộc sử dụng đệ quy.

Prototype:

```
void solution_ex1(int number)
```

Ví dụ: $solution\_ex1(100) \Rightarrow$ sẽ in ra màn hình 2 2 5 5.

Câu 2 *(4 điểm)*

Hai ma trận được gọi là tương thích với nhau khi tổng của 2 chúng tạo thành 1 ma trận mới có tất cả các phần tử bằng nhau. Cho ma trận A kích thước $m \times n$ và ma trận B kích thước $p \times q$ ($p < m$, $q < n$). Viết hàm tìm ma trận 3 chiều chứa tất cả các ma trận con của A tương thích với ma trận B.
Prototype:

```
int*** solution_ex2(int** matrixA, int m, int n, int** matrixB, int p, int q,
                                                 int& d1, int& d2, int& d3)
```

Ví dụ:

```
A = {{1, 2, 3}, {4, 5, 6}, {7, 8, 2}}, m = 3, n = 3.
B = {{4, 3}, {1, 0}}, p = 2, q = 2,
=> solution_ex2(A, 3, 3, B, 2, 2, d1, d2, d3)
    = {{{1, 2}, {4, 5}}, {{2, 3}, {5, 6}}, {{4, 5}, {7, 8}}}
```

với `d1 = 3, d2 = 2, d3 = 2`

Câu 3 *(4 điểm)*

Cho định nghĩa `Node` và `List` như dưới đây.

```
struct Node {
    char base; // Nucleotide base: A, T, G, X
    Node *next;
};
```

```
struct List { // 1 doan gene
    Node *head;
    Node *tail;
};
```

Với mỗi `Node` được xem như 1 nucleotide được biểu diễn bởi 1 trong 4 ký tự `{A, T, X, G}` và mỗi `List` là một đoạn gene chứa nhiều nucleotide. Biết rằng đoạn mã gene sẽ bị nhiễm độc khi một đoạn mã, có tên là `toxic_gene` xuất hiện. Hãy tìm và xóa đoạn mã `toxic_gene` **đầu tiên** khỏi đoạn mã gene cho trước.
Prototype:

```
void solution_ex3(List& gene, List toxic_gene)
```

Ví dụ:

```
gene = {A, A, X, X, G, X, T, A, T, X, G, X, X}
toxic_gene = {X, G, X}
=> gene (new) = {A, A, X, T, A, T, X, G, X, X}
```

Hết