**BÀI TẬP TUẦN 3 - TÌM KIẾM VÀ SẮP XẾP**

**Môn học: Cấu trúc dữ liệu và giải thuật - 21CTT4 - A&B**

Sinh viên lưu ý: nộp bài theo đúng quy định trong file “Quy định môn học lập trình C++”

Mục tiêu bài tập: giúp sinh viên rèn luyện khả năng lập trình C++ để xử lý các bài toán về tìm kiếm (search) và sắp xếp (sort).

1. Viết chương trình nhập vào mảng số nguyên gồm n phần tử và số nguyên x. Kiểm tra xem giá trị x có nằm trong mảng vừa nhập hay không.

Nếu x nằm trong mảng, in ra màn hình “exist”. Ngược lại, in ra màn hình “non-exist”.

| Std in | Std out |
| --- | --- |
| 5  1 2 3 4 5  10 | non-exist |
| 5  10 20 30 40 50  30 | exist |

1. Viết chương trình, nhập vào hai chuỗi s\_1 và s\_2. Kiểm tra xe, chuỗi s\_2 có nằm bên trong chuỗi s\_1 hay không.

Nếu chuỗi s\_2 nằm bên trong chuỗi s\_1, in ra màn hình “exist”. Ngược lại, in ra màn hình “non-exist”.

Lưu ý, không được dùng các hàm hỗ trợ của thư viện string để kiểm tra. Sinh viên phải xem hai chuỗi s\_1 và s\_2 như là hai mảng và xử lý trên hai mảng đó. Có phân biệt giữa ký tự in hoa và in thường.

| Std in | Std out |
| --- | --- |
| hello world  world | exist |
| university of science  social | non-exist |

1. Viết chương trình, nhập vào mảng số nguyên gồm n phần tử “đã được sắp xếp từ trước” và giá trị x. Viết chương trình kiểm tra xem phần tử x có nằm trong mảng vừa nhập hay không. Sử dụng binary search.

Nếu x nằm trong mảng, in ra màn hình “exist”. Ngược lại, in ra màn hình “non-exist”.

| Std in | Std out |
| --- | --- |
| 5  10 20 30 40 50  40 | exist |
| 5  10 20 30 40 50  100 | non-exist |

1. Viết chương trình, nhập vào mảng số nguyên gồm n phần tử và giá trị x. Đếm số lần xuất hiện của giá trị x trong mảng vừa nhập.

| Std in | Std out |
| --- | --- |
| 5  10 50 20 30 30  100 | 0 |
| 5  10 50 20 30 30  30 | 2 |

1. Viết chương trình, nhập vào mảng số nguyên gồm n phần tử. In ra màn hình, phần tử xuất hiện nhiều nhất trong mảng. Nếu có nhiều hơn hai phần tử là kết quả, thì in ra tất cả các phần tử đó.

| Std in | Std out |
| --- | --- |
| 5  10 50 20 30 30 | 30 |
| 7  10 50 20 30 30 50 50 | 50 |

1. Bubble sort
   1. Viết chương trình nhập vào mảng số nguyên gồm n phần tử. In ra màn hình các phần tử trong mảng đã được sắp xếp theo thứ tự từ bé đến lớn. Sinh viên sử dụng Bubble sort.
   2. Sinh viên cho biết độ phức tạp của thuật toán này và giải thích tại sao (ngắn gọn), bằng cách comment trong code.

| Std in | Std out |
| --- | --- |
| 5  10 50 20 30 40 | 10 20 30 40 50 |

1. Nội dung tương tự câu hỏi số 6. Nhưng với thuật toán Quick sort.
2. Viết chương trình nhập vào hai mảng số nguyên arr\_1 và arr\_2 đã được sắp xếp từ trước. In ra màn hình, mảng kết quả là ghép hai mảng vừa nhập. Lưu ý, mảng kết quả vẫn phải được sắp xếp từ bé đến lớn.

Sinh viên sử dụng ý tưởng của thuật toán Merge sort.

| Std in | Std out |
| --- | --- |
| 5  10 20 30 40 50  2  25 100 | 10 20 25 30 40 50 100 |
| 2  5 20  2  10 15 | 5 10 15 20 |

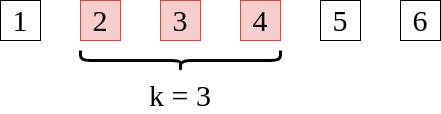
1. Viết chương trình nhập vào mảng số nguyên gồm n phần tử. Tìm tổng của hai phần tử trong mảng, sao cho tổng này có giá trị lớn nhất. In ra màn hình giá trị tổng đó.

Phân tích độ phức tạp của giải thuật của bạn và giải thích kết quả (ngắn gọn) bằng cách comment trong code.

| Std in | Std out |
| --- | --- |
| 5  10 50 20 30 40 | 90 |
| 5  -10 0 -20 -15 0 | 0 |

1. Viết chương trình nhập vào mảng số nguyên gồm n phần tử. Viết chương trình tìm mảng con gồm k phần tử liên tiếp (k < n), sao cho tổng của mảng con này là lớn nhất. Nếu có nhiều mảng con cùng có giá trị, in ra tất cả các mảng con đó.

Minh họa mảng con gồm k phần tử liên tiếp:



| Std in | Std out |
| --- | --- |
| 5  10 50 20 30 40  3 | 50 20 30 |
| 5  -10 0 6 -1 -5  3 | 0 6 -1 |