**Bài tập cấu trúc dữ liệu và giải thuật**

1. **Cơ bản về C++**

**Bài 1.** Dùng hàm cin, cout để nhập vào 3 số nguyên kiểm tra xem nó có thể tạo thành 3 cạnh của tam giác vuông không?

**Bài 2.** Nhập vào xâu ký tự (kiểu string) bằng cin xuất ra xâu viết hoa và xâu viết thường của xâu đó

**Bài 3.** Viết hàm tìm max và min của 3 số nguyên theo nguyên mẫu dưới đây, sau đó nhập vào 3 số nguyên tìm giá trị lớn nhất và nhỏ nhất

**void maxmin(int a,int b,int c,int &Max, int &Min);**

**Bài 4.** Viết hàm có đối mặc định để có thể vừa tìm max vừa tìm min tùy theo đối ismax theo nguyên mẫu dưới đây, sau đó áp dụng nhập vào 3 số nguyên khác nhau tìm số không phải max, min bằng cách tính tổng rồi trừ đi max và min

**int Sub(int a,int b,bool ismax=true);**

**Bài 5.** Viết hàm mẫu (template) tìm max của hai số, áp dụng tìm max của 2 số nguyên và hai số thực.

**Bài 6.** Viết hàm mẫu (template) tìm ước chung lớn nhất của 2 số, và hàm mẫu tìm ước chung lớn nhất của một dãy, áp dụng nhập vào một dãy kiểu long tìm ước chung lớn nhất của dãy.

1. **Lập trình hướng đối tượng trong C++**

**Bài 7.** Khai báo lớp điểm gồm 2 tọa độ và hàm nhập, và lớp đa giác gồm số đỉnh và con trỏ điểm, có các phương thức nhập các đỉnh, tính chu vi, tính điện tích đa giác. Viết hàm main mô phỏng chương trình

**Bài 8.** Khai báo lớp đa thức với bậc và mảng các hệ số, viết các phương thức nhập và xuất, tính giá trị của đa thức tại giá trị t. Viết hàm main thực hiện chương trình

**Bài 9.** Khai báo lớp ma trận nguyên vuông gồm cỡ ma trận và con trỏ 2 chiều chứa các phần tử của ma trận (int \*\*a), xây dựng các phương thức nhập, xuất, chuyển vị, kiểm tra tam giác trên, tính tổng, tìm phần tử lớn nhất. Viết hàm main mô phỏng

**Bài 10.** Khai báo lớp điểm có hàm tạo không đối gán tọa độ (0,0) và hàm tọa có đối gán tọa độ (a,b), phương thức xuất, tính diện tích hình tròn có đường kính đi qua gốc tọa độ và điểm hiện tại. Viết hàm main mô phỏng

**Bài 11**. Khai báo lớp đa thức có hệ số là con trỏ chưa cấp phát (float \*a) viết hàm tạo có đối số là bậc của đa thức, cấp phát bộ nhớ động, viết hàm hủy giải phóng bộ nhớ động, phương thức nhập, xuất đa thức.

**Bài 12.** Xây dựng lớp biểu diễn các đối tượng dãy số với các phương thức hàm tạo, hàm in, hàm thêm một phần tử vào dãy, hàm xóa một phần tử của dãy, hàm tìm kiếm một phần tử có trong dãy không nếu có trả lại vị trí của phần tử đó trong dãy.

**Bài 13.** Xây dựng lớp biểu diễn các đối tượng là các sinh viên (các thuộc tính, phương thức do sv tự xác định)

**Bài 14.** Xây dựng lớp biểu diễn tập hợp số nguyên có hàm tạo, hàm hủy, có hàm rút gọn những phần tử trùng nhau chỉ giữ lại 1 phần tử, hàm sắp xếp tăng dần

**Bài 15.** Xây dựng lớp mẫu phân số thực hiện nhập vào n số nguyên a1,... an tìm giá trị tối giản của phân số



**Bài 16.** Xây dựng lớp mẫu số phức thực hiện nhập vào n số phức tính tổng, hiệu, tích bằng phương thức toán tử

**Bài 17.** Xây dựng lớp biểu diễn các vector trong không gian n chiều có các phương thức toán tử: +, - hai vector, \* tích vô hướng hai véc tơ,-(đổi dấu) >>, <<.

**Bài 18.** Xây dựng lớp biểu diễn các đa thức với các phương thức toán tử: +, -, \* hai đa thức, tính giá trị đa thức, >>, <<.

**Bài 19.** Xây dựng lớp biểu diễn các ma trận có các phương thức toán tử: +, - ,\* hai ma trận, >>, <<.

1. **Phân tích độ phức tạp**

**Bài 20.** Cho hàm sau phân tích độ phức tạp về thời gian

|  |
| --- |
| **int** sum**(int** n**)**  **{**  **int** s**=**1**;**  **for(int** i**=**n**;**i**>**0**;**i**-=**3**)** s**\*=**i**;**  **return** s**;**  **}** |

**Bài 21.** Cho hàm sau phân tích độ phức tạp về thời gian

|  |
| --- |
| **void** sort**(int** n**,int** **\***a**,bool** ok**=true)**  **{**  **for(int** i**=**0**;**i**<**n**;**i**++)**  **for(int** j**=**i**+**1**;**j**<**n**;**j**++)**  **if(**a**[**i**]>**a**[**j**]==**ok**)**  **{**  **int** t**=**a**[**i**];**  a**[**i**]=**a**[**j**];**  a**[**j**]=**t**;**  **}**  **}** |

**Bài 22.** Cho hàm sau phân tích độ phức tạp về thời gian

|  |
| --- |
| **int** fibo**(int** n**)**  **{**  **int** f**[**n**+**5**];**  f**[**0**]=**f**[**1**]=**1**;**  **for(int** i**=**2**;**i**<**n**;**i**++)** f**[**i**]=**f**[**i**-**1**]+**f**[**i**-**2**];**  **return** f**[**n**];**  **}** |

**Bài 23.** Cho hàm sau phân tích độ phức tạp về thời gian

|  |
| --- |
| **int** TimKiemNhiPhan**(int** n**,** **int\***a**,** **int** x**){**  **int** L **=**1**,** R **=**n **;**  **while(**L**<=**R**)**  **{**  **int** M **=** **(**L **+** R**)/**2**;**  **if(**x**==**a**[**M**])** **return** M**;**  **if(**x **<** a**[**M**]** **)** R**=** M **-** 1**;**  **else** L **=** M **+** 1**;**  **}**  **return** 0**;**  **}** |

**Bài 23.** Cho hàm sau phân tích độ phức tạp về thời gian

|  |
| --- |
| **bool** isprime**(int** n**)**  **{**  **if(**n**==**2**)** **return** **true;**  **if(**n**<**2 **||** **(**n**%**2**==**0**))** **return** **false;**  **for(int** i**=**3**;**i**\***i**<=**n**;**i**+=**2**)**  **if(**n**%**i**==**0**)** **return** **false;**  **return** **true;**  **}** |

1. **Đệ quy**

**Bài 24.** Viết hàm đệ quy tính n! và hàm main mô phỏng

**Bài 25.** Viết hàm đệ quy tính tổ hợp chập k của n và hàm main mô phỏng

**Bài 26.** Viết hàm đệ quy in ra màn hình tam giác \*

**Bài 27.** Viết hàm đệ quy tính tổng của dãy a1+a2+...+an

**Bài 28.** Viết hàm đệ quy tính số fibonacci có sử dụng một mảng phụ trợ nếu đã tính rồi thì tra trong mảng nếu chưa có mới tính

**Bài 29.** Viết hàm đệ quy tương hỗ tính sinx và cosx với sai số epsilon theo công thức





**Bài 30.** Viết hàm đệ quy mô phỏng chuyển Tháp Hà Nội

1. **Cấu trúc dữ liệu tuyến tính**

**Bài 31.** Áp dụng vector trong thư viện STL xây dựng lớp đa thức thực hiện tính tổng hai đa thức.

**Bài 32.** Áp dụng vector trong thư viện STL xây dựng lớp số lớn để giải bài toán tính giai thừa tới tối đa 1000!

**Bài 33**. Áp dụngvector trong thư viện STL xây dựng lớp lớp học gồm n sinh viên và các sinh viên có thông tin về tên, tuổi, điểm. Nhập danh sách n sinh viên tính điểm trung bình, sắp xếp danh sách theo điểm giảm dần.

**Bài 34.** Áp dụngvector trong thư viện STL xây dựng lớp ma trận tìm ma trận chuyển vị

**Bài 34.** Áp dụngvector trong thư viện STL xây dựng lớp tập hợp cho phép rút gọn tập hợp, tìm hợp, giao của hai tập hợp.

**Bài 35.** Cài đặt cấu trúc vector trừu tượng bổ sung thêm phương thức sắp xếp tăng dần các phần tử, phương thức tìm max, min.

1. **Cấu trúc dữ liệu phi tuyến**