Đề thi
KỸ THUẬT LẬP TRÌNH
(120 phút, không dùng tài liệu)

Câu 1

Một chương trình đơn giản để quản lý thông tin sinh viên, mỗi sinh viên lưu các thông tin sau:

- Mã số sinh viên (chuỗi gồm tối đa 10 ký tự)
- Họ và tên (chuỗi ký tự có kích thước biến động)
- Năm sinh (số nguyên dương)
- Khóa tuyển (chuỗi gồm tối đa 12 ký tự)
- Điểm trung bình (số thực, không âm)

Yêu cầu: Hãy viết khai báo dữ liệu có cấu trúc để lưu trữ thông tin của mỗi sinh viên.

Câu 2

Kiểu *vector<T>* của thư viện chuẩn *STL* được hướng dẫn sử dụng trong bảng sau:

Phương thức	Ý nghĩa
`size( );`	Trả về kích thước hiện hành của mảng.
`resize(int newsize);`	Thay đổi kích thước mảng, nếu mảng co lại (kích thước mới nhỏ hơn cũ) thì một số phần tử bị xóa khỏi mảng.
`push_back(T x);`	Thêm phần tử **x** (có kiểu **T** vào cuối mảng), mảng tự động giãn ra.
`pop_back( );`	Xóa phần tử cuối cùng của mảng, mảng tự động giảm kích thước bớt 1 phần tử.

Người ta sử dụng kiểu *vector<T>* để cài đặt một hàng đợi gồm các số nguyên. Mã nguồn được viết trước một phần như sau.

```
#include <vector>
using namespace std;

typedef struct {
        vector<int> data;
} IntQueue;

void enqueue(IntQueue& q, int x);
int dequeue(IntQueue& q);
int first(IntQueue& q);
```

Yêu cầu: Hãy viết mã nguồn đầy đủ cho các hàm *enqueue( )*, *dequeue( )*, *first( )*.

Trong trường hợp thực sự cảm thấy cần thiết, bạn có thể sửa đổi cấu trúc *IntQueue* cho phù hợp.

Câu 3

Hàm nhập chuỗi ký tự sau đây còn ***bị một lỗi lập trình có thể nói khá là nghiêm trọng***, một số trình biên dịch không thể báo lỗi, mà ngay cả khi chạy chương trình cũng có lúc vẫn chạy có vẻ như là đúng.

```
1  #include <stdio.h>
2  const int MaxSize = 1024;
3  char* strInput( ) // Hàm viết sai
4  {
5          char Buff[MaxSize];
6          scanf("%s", Buff);
7          return &(Buff[0]);
8  }
```

Yêu cầu: Bạn hãy tìm ra lỗi của hàm nói trên và viết lại mã của hàm một cách chính xác, với điều kiện là vẫn phải giữ nguyên khai báo hàm là: **char*** strInput( ).

Câu 4

Hàm sắp xếp các số thực theo thứ tự tăng dần được khai báo như sau:

```
void Sort(float a[], int n) ;
```

Yêu cầu: Hãy viết mã nguồn cho hàm này với điều kiện không được sử dụng bất kỳ một cấu trúc lặp nào (tức là không dùng vòng lặp để lập trình cho câu hỏi này).

Tự chọn. Bạn làm tiếp tục đề thi này bằng cách chọn một trong hai bài sau đây.

Câu 5A

Một số hoàn chỉnh là số nguyên dương bằng tổng các ước số nhỏ hơn nó. Chẳng hạn: 6 và 28 là hai số hoàn chỉnh bởi vì: $6 = 1 + 2 + 3$ và $28 = 1 + 2 + 4 + 7 + 14$.

a) **Yêu cầu**: Viết hàm để kiểm tra xem một số nguyên dương ***n*** có là số hoàn chỉnh hay không.

b) Các nhà toán học đã chứng minh rằng một số nguyên dương ***n*** là số hoàn chỉnh chẵn khi và chỉ khi $n = 2^m(2^m - 1)$ với $2^m - 1$ là số nguyên tố. Nhờ vậy việc kiểm tra một số chẵn bất kỳ có là số hoàn chỉnh hay không có thể được một cách rất nhanh chóng ngay cả khi ***n*** rất lớn.

Yêu cầu: Dựa vào tính chất trên, bạn hãy đề xuất thuật toán để kiểm tra một số chẵn bất kỳ có là số hoàn chỉnh hay không. Bạn có thể viết một phần mã nguồn minh họa.

Câu 5B

Tập tin nhập: là tập tin nhị phân lưu trữ các số thực kiểu ***float***, mỗi số lưu trữ 4 byte, các số được lưu trữ liền kề nhau. Chú ý là đầu tập tin ***không có*** lưu trữ số lượng các số.

Tập tin xuất: là tập tin nhị phân, đầu tập tin lưu số nguyên 2 byte để lưu số lượng các số thực kiểu ***float*** (mỗi số lưu trữ 4 byte) lưu tiếp theo sau và liền kề nhau trong tập tin. Chẳng hạn nếu có 10 số thực thì tập tin có kích thước: 2 + 10 x 4 = 42 byte, trong đó 2 byte đầu tiên lưu số 10, phần còn lại của tập tin lưu liên tiếp 10 số, mỗi số chiếm 4 byte.

Yêu cầu: Viết hàm đọc tập tin nhập và ghi vào tập tin xuất sao cho các số trong tập tin xuất được sắp theo thứ tự trị tuyệt đối giảm dần.

------ HẾT -----