

CÂU HỎI MÔN TIN HỌC CƠ SỞ 2

(Lập trình và viết lưu đồ)

Không được sử dụng các hàm trong string.h

Loại bài mức A

1. Đổi số nguyên thành số nhị phân. Dữ liệu đọc từ tệp.
char *doinhiphan(long n);
void doctep(long &n);
2. Sắp xếp dãy các số thực a[0], a[1],..., a[n-1] theo thứ tự tăng dần bằng giải thuật chọn Selection Sort. Dữ liệu đọc ghi trên tệp.
void selectsort(float a[], int n);
void doctep(float a[], int &n);
void ghitep(float a[], int n);
3. Sắp xếp dãy các số thực a[0], a[1],..., a[n-1] theo thứ tự giảm dần bằng giải thuật chèn Insertion Sort. Dữ liệu đọc ghi trên tệp.
void insertionsort(float a[], int n);
void doctep(float a[], int &n);
void ghitep(float a[], int n);
4. . Sắp xếp dãy các số thực a[0], a[1],..., a[n-1] theo thứ tự giảm dần bằng giải thuật nổi bọt Bubble Sort. Dữ liệu đọc ghi trên tệp.
void bubblesort(float a[], int n);
void doctep(float a[], int &n);
void ghitep(float a[], int n);
5. Tính n! Dữ liệu đọc từ tệp.
long giaithua(int n);
void doctep(int n);
6. Đếm số lần xuất hiện của số x trong dãy số thực a[0], a[1],..., a[n-1]. Dữ liệu đọc trên tệp.
int demso(float a[], int n, float x);
void doctep(float a[], int &n, float &x);
7. Tính tổng các số dương trong dãy số thực a[0], a[1],..., a[n-1]. Dữ liệu đọc trên tệp.
float tongduong(float a[], int n);
void doctep(float a[], int &n);
8. Tính tổng các số âm trong dãy số thực a[0], a[1],..., a[n-1]. Dữ liệu đọc trên tệp.
float tongam(float a[], int n);
void doctep(float a[], int &n);
9. Tính giá trị lớn nhất trong dãy số thực a[0], a[1],..., a[n-1]. Dữ liệu đọc trên tệp.
float lonnhat(float a[], int n);
void doctep(float a[], int &n);

10. Tính giá trị nhỏ nhất trong dãy số thực $a[0], a[1], \dots, a[n-1]$. Dữ liệu đọc trên tệp.

```
float nhonhat(float a[], int n);
void doctep(float a[], int &n);
```
11. Đếm tổng số ký tự là chữ hoa trong một xâu s[]. Dữ liệu đọc từ tệp.

```
int tongchuhua(char s[]);
void doctep(char s[]);
```
12. Đếm tổng số ký tự là chữ thường trong một xâu s[]. Dữ liệu đọc từ tệp

```
int tongchuthuong(char s[]);
void doctep(char s[]);
```
13. Đổi một xâu s[] từ chữ thường thành chữ hoa. Dữ liệu đọc từ tệp

```
char *doithuhua(char s[]);
void doctep(char s[]);
```
14. Đổi một xâu s[] từ chữ hoa thành chữ thường. Dữ liệu đọc từ tệp

```
char *doichuthuong(char s[]);
void doctep(char s[]);
```
15. Kiểm tra dãy số thực $a[0], a[1], \dots, a[n-1]$ đã được sắp xếp theo thứ tự tăng dần hay không. Dữ liệu đọc trên tệp.

```
int kiemtrasapxep(float a[], int n);
void doctep(float a[], int &n);
```
16. Đếm tổng số ký tự là chữ số trong một xâu s[]. Dữ liệu đọc từ tệp

```
int tongchuso(char s[]);
void doctep(char s[]);
```

Loại bài mức B

1. Đếm số từ trong một xâu ký tự. Thí dụ chuỗi "Trường học " có 2 từ. Dữ liệu đọc từ tệp

```
int demtu(char s[]);
void doctep(char s[]);
```
2. Đếm số lần xuất hiện của các ký tự 'a', 'b', 'c', ... trong xâu s[], có phân biệt chữ hoa chữ thường. Dữ liệu đọc ghi trên tệp.

```
void demkytu1(char s[], char kt[], int sl[], int &tskt);
void doctep(char s[]);
void ghitep(char kt[], int sl[], int tskt);
```
3. Đếm số lần xuất hiện của các ký tự 'a', 'b', 'c', ... trong xâu s[], không phân biệt chữ hoa chữ thường Dữ liệu đọc ghi trên tệp.

```
void demkytu2(char s[], char kt[], int sl[], int &tskt);
void doctep(char s[]);
void ghitep(char kt[], int sl[], int tskt);
```
4. Kiểm tra xâu s1[] có chứa xâu s2[] hay không. Dữ liệu đọc từ tệp.

```
int ktxau(char s1[], char s2[]);
void doctep(char s1[], char s2[]);
```
5. Trích n ký tự bên trái của một xâu s[] từ vị trí m. Dữ liệu đọc từ tệp.

- ```
char *trichtraib(char s[],int n, int m);
void doctep(char s[],int &n, int &m);
```
6. Trích n ký tự bên phải một xâu s[]. Dữ liệu đọc từ tệp
- ```
char *trichphai(char s[], int n);
void doctep(char s[], int &n);
```
7. Nhập một số $c > 0$ (ví dụ $c = 0.0001$) và một số thực x rồi tính
- $$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$
- tổng được tính với n đủ lớn sao cho bất đẳng thức $\left| \frac{x^n}{n!} \right| \leq c$ thỏa mãn.
- ```
float ex(float x, float c);
```
8. Nhập số liệu cho dãy số thực  $a[0], a[1], \dots, a[n-1]$  và một giá trị thực  $x$ . Giả sử dãy  $a[]$  đã được sắp xếp theo thứ tự tăng dần. Hãy chèn giá trị  $x$  vào dãy sao cho dãy  $a[]$  vẫn tăng. Dữ liệu đọc ghi trên tệp.
- ```
void chenx(float a[], int &n, float x);
void doctep(float a[], int &n, float &x);
void ghitep(float a[], int n);
```
9. Tính giá trị lớn nhất và các vị trí của nó trong dãy số thực $a[0], a[1], \dots, a[n-1]$. Dữ liệu đọc ghi trên tệp.
- ```
float lonnhat(float a[], int n, int vitri[], int &tongvitri);
void doctep(float a[], int &n);
void ghitep(int vitri[], int tongvitri, float max);
```
10. Tính giá trị nhỏ nhất và các vị trí của nó trong dãy số thực  $a[0], a[1], \dots, a[n-1]$ . Dữ liệu đọc ghi trên tệp.
- ```
float nhonhat(float a[], int n, int vitri[], int &tongvitri);
void doctep(float a[], int &n);
void ghitep(int vitri[], int tongvitri, float min);
```
11. Hãy liệt kê các phần tử xuất hiện trong dãy số thực $a[0], a[1], \dots, a[n-1]$ đúng một lần. Dữ liệu đọc ghi trên tệp.
- ```
void solan1(float a[], int n, float a1[], int &m);
void doctep(float a[], int &n);
void ghitep(float a1[], int m);
```
12. Hãy liệt kê các phần tử xuất hiện trong dãy số thực  $a[0], a[1], \dots, a[n-1]$  đúng hai lần. Dữ liệu đọc ghi trên tệp.
- ```
void solan2(float a[], int n, float a2[], int &m);
void doctep(float a[], int &n);
void ghitep(float a2[], int m);
```

Loại bài mức C

1. Cho 2 chuỗi $s1[]$ và $s2[]$ nhập từ tệp. Hãy tìm xem chuỗi $s1$ có chứa chuỗi $s2$ không và chỉ rõ vị trí bắt đầu và vị trí kết thúc của chuỗi $s2$ trong chuỗi $s1$ nếu tìm thấy.
- ```
void doctep(char s1[], char s2[]);
```

- int timxau(char s1[], char s2[], int &vt dau, int &vt cuoi);
2. Cho 2 chuỗi s1[] và s2[] nhập từ tệp. Hãy tìm s1[] trong s2[] xóa s1[] trong s2[] nếu tìm thấy.  
void doctep(char s1[], char s2[]);  
void xoatu(char s1[], char s2[]);
  3. Nhập số liệu cho dãy số thực a[0], a[1],..., a[n-1] từ tệp. Tìm 2 số lớn nhất khác nhau và vị trí của chúng trong dãy trên (nếu có hai số cùng giá trị thì lấy chỉ số nhỏ hơn). Thí dụ trong dãy 1,5,3,4,5 thì 2 phần tử lớn nhất là 5 và 4 và ở các vị trí 1 và 3.  
void doctep(float a[], int &n);  
void timso(float a[], int n, float &max1, int &vt1, float &max2, int &vt2);
  4. Nhập số liệu cho dãy số thực a[0], a[1],..., a[n-1] từ tệp. Hãy liệt kê các phần tử lớn hơn tất cả các phần tử đứng trước nó. Kết quả ghi ra tệp.  
void doctep(float a[], int &n);  
void truoc(float a[], int n, float b1[], int &m);  
void ghitep(float b1[], int m);
  5. Nhập số liệu cho dãy số thực a[0], a[1],..., a[n-1] từ tệp. Hãy liệt kê các phần tử lớn hơn tất cả các phần tử đứng sau nó. Kết quả ghi ra tệp.  
void doctep(float a[], int &n);  
void sau(float a[], int n, float b2[], int &m);  
void ghitep(float b2[], int m);
  6. Nhập số liệu cho dãy số thực a[0], a[1],..., a[n-1] từ tệp. Đếm số lần xuất hiện các phần tử trong dãy.  
void doctep(float a[], int &n);  
void demso(float a[], int n, float so[], int sl[], int &tongso);  
void ghitep(float so[], int sl[], int &tongso);
  7. Nhập số n và dãy các số thực a[0], a[1],..., a[n-1] từ tệp, rồi sắp xếp dãy trên theo thứ tự tăng dần theo phương pháp nhanh (quick sort). Kết quả ghi ra tệp.  
void doctep(float a[], int &n);  
void quicksort(float a[], int n);  
void ghitep(float a[], int n);
  8. Xây dựng các thao tác sau với danh sách liên kết đơn có cấu trúc:

```

struct canbo { long maso; char ten[8]; };
struct node { canbo info; node *next; };
struct danh sach { node *pfirst, *plast; };

```

    1. Khởi tạo danh sách
    2. Kiểm tra danh sách rỗng.
    3. Thêm một phần tử vào đầu danh sách.
    4. Thêm một phần tử vào cuối danh sách.
    5. Thêm một phần tử vào sau vị trí thứ k.
    6. Thêm một phần tử vào trước vị trí thứ k

7. Xoá phần tử đầu danh sách.
8. Xoá phần tử cuối danh sách.
9. Xoá phần tử thứ k.
10. Xoá toàn bộ danh sách.
11. Xem danh sách trên màn hình
12. Đọc danh sách từ tệp.
13. Ghi danh sách vào tệp