CÂU HỎI ÔN TẬP MÔN TIN HỌC CƠ SỞ 2

(Lập trình và viết lưu đồ)

Loại bài mức A

1.Nhập một số nguyên dương n, sau đó đổi số đó dưới dạng nhị phân.
char *doinhiphan(long n);

2. Nhập số n và dãy các số thực a[0], a[1],..., a[n-1] rồi sắp xếp dãy trên theo thứ tự tăng dần bằng giải thuật chọn Selection Sort.
void selectsort(float a[], int n);

3. Nhập số n và dãy các số thực a[0], a[1],..., a[n-1] rồi sắp xếp dãy trên theo thứ tự tăng dần bằng giải thuật chèn Insertion Sort.
void insertionsort(float a[], int n);

4. Nhập số n và dãy các số thực a[0], a[1],..., a[n-1] rồi sắp xếp dãy trên theo thứ tự tăng dần bằng giải thuật nổi bọt Bubble Sort.
void bubblesort(float a[], int n);

5. Tính n!
long giaithua(int n);

6. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., a_{n-1} và một số x bất kỳ. Đếm số lần xuất hiện của số x trong dãy trên.
int demso(float a[], int n, float x);

7. Nhập số liệu cho dãy số thực a_0, a_1 ,..., a_{n-1}. Tính tổng các số dương trong dãy.
float tongduong(float a[], int n);

8. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., a_{n-1} . Tính tổng các số âm trong dãy.
float tongam(float a[], int n);

9. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., a_{n-1} . Tính giá trị lớn nhất trong dãy.
float lonnhat(float a[], int n);

10. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., a_{n-1} . Tính giá trị nhỏ nhất trong dãy.
float nhonhat(float a[], int n);

11. Đếm tổng số ký tự là chữ hoa trong một xâu
int tongchuhoa(char s[]);

12. Đếm tổng số ký tự là chữ thường trong một xâu
int tongchuthuong(char s[]);

13. Đổi một xâu từ chữ thường thành chữ hoa
char *doithuhoa(char s[]);

14. Đổi một xâu từ chữ hoa thành chữ thường
char *doichuthuong(char s[]);

15. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., a_{n-1} . Kiểm tra xem dãy đã được sắp xếp theo thứ tự tăng dần hay không.
int kiemtrasapxep(float a[], int n);

16. Đếm tổng số ký tự là chữ số trong một xâu
int tongchuso(char s[]);

Loại bài mức B

1. Đếm số từ trong một xâu ký tự. Thí dụ chuỗi "Trường học " có 2 từ.
   int demtu(char s[]);
2. Cho một chuỗi ký tự có độ dài n, hãy đếm số lần xuất hiện của các ký tự 'A','B','C' theo cách: Có phân biệt chữ hoa chữ thường.
   void demkytu1(char s[], char kt[],int sl[], int &tskt);
3. Cho một chuỗi ký tự có độ dài n, hãy đếm số lần xuất hiện của các ký tự 'A','B','C' theo cách: Không phân biệt chữ hoa chữ thường.
   void demkytu2(char s[], char kt[],int sl[], int &tskt);
4. Kiểm tra một xâu có chứa xâu khác hay không.
   int ktxau(char s1[], char s2[]);
5. Trích n ký tự bên trái của một xâu từ vị trí m
   char *trichtrai(char s[],int n);
6. Trích n ký tự bên phải một xâu
   char *trichphai(char s[], int n);
7. Nhập một số c>0 (ví dụ c = 0.0001) và một số thực x rồi tính

   $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + ... + \frac{x^n}{n!}$

   tổng được tính với n đủ lớn sao cho bất đẳng thức $|\frac{x^n}{n!}| \le c$ thỏa mãn.

   float ex(float x, float c);
8. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., 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 a sao cho vẫn giữ được tính sắp xếp của mảng.
   void chenx(float a[], int &n,float x)
9. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., a_{n-1} . Tính giá trị lớn nhất và các vị trí của nó trong dãy.
   float lonnhat(float a[], int n, int vitri[], int &tongvitri);
10. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., a_{n-1} . Tính giá trị nhỏ nhất và các vị trí của nó trong dãy.
    float nhonhat(float a[], int n, int vitri[], int &tongvitri);
11. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., a_{n-1} từ tệp. Hãy liệt kê các phần tử xuất hiện trong dãy đúng một lần.
    void solan1(float a[], int n, float a1[], int &m);
12. Nhập số liệu cho dãy số thực a_0 , a_1 ,..., a_{n-1} từ tệp. Hãy liệt kê các phần tử xuất hiện trong dãy đúng 2 lần.
    void solan2(float a[], int n, float a2[], int &m);

Loại bài mức C

1. Cho 2 chuỗi s1 và s2, chuỗi s1 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(float s1[]);
int timxau(char s1[], char s2[], int &vtdau, int &vtcuoi);

2. Cho một chuỗi s2 nhập từ tệp. Hãy tìm một từ s1 và xóa từ này trong chuỗi s2 nếu tìm thấy.
   void doctep(float 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.
   void doctep(float a[], int &n);
   void truoc(float a[], int n, 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.
   void doctep(float a[], int &n);
   void sau(float a[], int n, float b2[], int &m);
6. Nhập số liệu cho dãy số nguyên 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(int a[], int n, int 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).
   void doctep(float a[], int &n);
   void quicksort(float a[], int n);
8. 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 giảm dần theo phương pháp nhanh (quick sort).
   void doctep(float a[], int &n);
   void quicksort(float a[], int n);
9. Xây dựng các thao tác sau với danh sách liên kết đơn có cấu trúc:

```
struct canbo {      maso; char ten[8]; };
struct node {       canbo info; node *next; };
struct danhsach {   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