

## CÂU HỎI MÔN TIN HỌC CƠ SỞ 2

(Lập trình bằng C hoặc C++ và viết lưu đồ)

Không được sử dụng các hàm trong string.h và ctype.h

1. Đổi số nguyên thành số nhị phân. Dữ liệu đọc từ tệp. Lập trình với các hàm:

char \*doinhiphan(long n);

void doctep(long &n);

2. Đổi số lẻ thành số nhị phân. Dữ liệu đọc từ tệp. Lập trình với các hàm:

char \*doinhiphan(float x, int n);

void doctep(float &x, int &n);

3. Sắp xếp dãy các số thực  $a[0], a[1], \dots, a[n-1]$  theo thứ tự tăng dần bằng giải thuật chọn Selection Sort. Dữ liệu đọc ghi trên tệp. Lập trình với các hàm:

void selectsort(float a[], int n);

void doctep(float a[], int &n);

void ghitep(float a[], int n);

*Ghi chú: Selection Sort là thuật toán sắp xếp thứ tự lần lượt từ trái qua phải hoặc ngược lại.*

*Bước 1: Thiết lập MIN về vị trí 0*

*Bước 2: Tìm kiếm phần tử nhỏ nhất trong danh sách*

*Bước 3: Trao đổi với giá trị tại vị trí MIN*

*Bước 4: Tăng MIN để trở tới phần tử tiếp theo*

*Bước 5: Lặp lại cho tới khi toàn bộ danh sách đã được sắp xếp*

4. Sắp xếp dãy các số thực  $a[0], a[1], \dots, a[n-1]$  theo thứ tự giảm dần bằng giải thuật chèn Insertion Sort. Dữ liệu đọc ghi trên tệp. Lập trình với các hàm:

void insertionsort(float a[], int n);

void doctep(float a[], int &n);

void ghitep(float a[], int n);

*Ghi chú: Insertion Sort là thuật toán sắp xếp phần tử đã kiểm tra chèn vào giữa dãy đã sắp xếp.*

*Bước 1: Kiểm tra nếu phần tử đầu tiên đã được sắp xếp trả về 1*

*Bước 2: Lấy phần tử kế tiếp*

*Bước 3: So sánh với tất cả phần tử trong danh sách con đã qua sắp xếp*

*Bước 4: Dịch chuyển tất cả phần tử trong danh sách con mà lớn hơn giá trị để được sắp xếp*

*Bước 5: Chèn giá trị đó*

*Bước 6: Lặp lại cho tới khi danh sách được sắp xếp*

5. Sắp xếp dãy các số thực  $a[0], a[1], \dots, a[n-1]$  theo thứ tự giảm dần bằng giải thuật nổi bọt Bubble Sort. Dữ liệu đọc ghi trên tệp. Lập trình với các hàm:

void bubblesort(float a[], int n);

void doctep(float a[], int &n);

void ghitep(float a[], int n);

*Ghi chú: Thuật toán Bubble Sort là hoán vị liên tục 2 phần tử liền kề nhau.*

6. Tính  $n!$  Dữ liệu đọc từ tệp. Lập trình với các hàm:

```
long giaithua(int n);
```

```
void doctep(int &n);
```

7. Đếm số lần xuất hiện của số  $x$  trong dãy số thực  $a[0], a[1], \dots, a[n-1]$ . Dữ liệu đọc trên tệp. Lập trình với các hàm:

```
int demso(float a[], int n, float x);
```

```
void doctep(float a[], int &n, float &x);
```

8. Tính tổng các số dương trong dãy số thực  $a[0], a[1], \dots, a[n-1]$ . Dữ liệu đọc trên tệp. Lập trình với các hàm:

```
float tongduong(float a[], int n);
```

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

9. Tính tổng các số âm trong dãy số thực  $a[0], a[1], \dots, a[n-1]$ . Dữ liệu đọc trên tệp. Lập trình với các hàm:

```
float tongam(float a[], int n);
```

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

10. Tính giá trị lớn nhất trong dãy số thực  $a[0], a[1], \dots, a[n-1]$ . Dữ liệu đọc trên tệp. Lập trình với các hàm:

```
float lonnhat(float a[], int n);
```

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

11. Tính giá trị nhỏ nhất trong dãy số thực  $a[0], a[1], \dots, a[n-1]$ . Dữ liệu đọc trên tệp. Lập trình với các hàm:

```
float nhonhat(float a[], int n);
```

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

12. Đếm tổng số ký tự là chữ hoa trong một xâu  $s[]$ . Dữ liệu đọc từ tệp. Lập trình với các hàm:

```
int tongchuhua(char s[]);
```

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

13. Đếm tổng số ký tự là chữ thường trong một xâu  $s[]$ . Dữ liệu đọc từ tệp. Lập trình với các hàm:

```
int tongchuthuong(char s[]);
```

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

14. Đổi một xâu  $s[]$  từ chữ thường thành chữ hoa. Dữ liệu đọc từ tệp. Lập trình với các hàm:

```
char *doichuhua(char s[]);
```

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

15. Đổi một xâu  $s[]$  từ chữ hoa thành chữ thường. Dữ liệu đọc từ tệp. Lập trình với các hàm:

```
char *doichuthuong(char s[]);
```

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

16. 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. Lập trình với các hàm:

- ```

    int kiemtrasapxep(float a[], int n);
    void doctep(float a[], int &n);

```
17. Đếm tổng số ký tự là chữ số trong một xâu s[]. Dữ liệu đọc từ tệp. Lập trình với các hàm:
- ```

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

```
18. 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);

```
19. 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[]);

```
20. 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);

```
21. 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);

```
22. Nhập số liệu cho dãy số thực  $a[0], a[1], \dots, a[n-1]$  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);

```
23. 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[]);

```
24. Trích n ký tự bên trái của một xâu s[] từ vị trí m. Dữ liệu đọc từ tệp.
- ```

    char *trichtrai(char s[], int n, int m);
    void doctep(char s[], int &n, int &m);

```
25. 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);

```

26. 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);
```
27. 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);
```
28. 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);
```
29. 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);
```
30. 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);
```
31. 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);
```
32. 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[]);
```
33. Nhập số liệu cho dãy số thực  $a[0], a[1], \dots, a[n-1]$  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);
```

34. Nhập số  $n$  và dãy các số thực  $a[0], a[1], \dots, a[n-1]$  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);
```

35. Nhập số  $n$  và dãy các số thực  $a[0], a[1], \dots, a[n-1]$  từ tệp, rồi sắp xếp dãy trên theo thứ tự tăng dần theo phương pháp trộn (merge sort). Kết quả ghi ra tệp.

```
void doctep(float a[], int &n);  
void mergesort(float a[], int n);  
void ghitep(float a[], int n);
```

36. **Xây dựng các thao tác sau với danh sách liên kết đơn**

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

1. Khởi tạo danh sách:  
void khoitao(danhsach &d);
2. Kiểm tra danh sách rỗng:  
int ktrong(danhsach d);
3. Thêm một phần tử vào đầu danh sách:  
void themdau(danhsach &d, canbo x)
4. Thêm một phần tử vào cuối danh sách.  
void themcuoi(danhsach &d, canbo x)
5. Thêm một phần tử vào sau vị trí thứ  $k$ .  
void themsau(danhsach &d, canbo x, int k)
6. Thêm một phần tử vào trước vị trí thứ  $k$   
void themtruock(danhsach &d, canbo x, int k)
7. Xoá phần tử đầu danh sách.  
void xoadau(danhsach &d)
8. Xoá phần tử cuối danh sách.  
void xoacuoi(danhsach &d)
9. Xoá phần tử thứ  $k$ .  
void xoak(danhsach &d, int k)
10. Xoá toàn bộ danh sách.  
void xoa(danhsach &d)
11. Xem danh sách trên màn hình  
void xem(danhsach d)
12. Đọc danh sách từ tệp.  
void doctep(danhsach &d)
13. Ghi danh sách vào tệp  
void ghitep(danhsach d)

**Giải thuật quay lui:**

**37.** Hãy viết chương trình liệt kê tất cả các số tự nhiên  $K[]$  thỏa mãn đồng thời những điều kiện dưới đây:

- (i)  $K$  là số có 5 chữ số;
- (ii)  $K$  là số nguyên tố;
- (iii)  $K$  là số thuận nghịch ( $k$  là số thuận nghịch nếu đọc xuôi hay đọc ngược các chữ số của  $k$  ta đều nhận được một số như nhau. Ví dụ số: 30303);
- (iv) *Biểu diễn của  $K$  ở hệ cơ số  $B$  ( $B$  bất kỳ được nhập từ bàn phím cũng là một số thuận nghịch. Ví dụ số  $k=30303$  có biểu diễn ở hệ cơ số 8 là 73137 cũng là một số thuận nghịch;*

**38.** Hãy viết chương trình liệt kê tất cả các số tự nhiên  $K[]$  thỏa mãn đồng thời những điều kiện dưới đây:

- (i)  $K$  là số có 5 chữ số;
- (ii)  $K$  là số nguyên tố;
- (iii) Đảo ngược các chữ số trong của  $K$  cũng là một số nguyên tố;
- (iv) Tổng các chữ số của  $K$  cũng là một số nguyên tố;
- (v) Mỗi chữ số trong  $K$  cũng là những số nguyên tố.

**39.** Hãy viết chương trình liệt kê tất cả các số tự nhiên  $K[]$  thỏa mãn đồng thời những điều kiện dưới đây:

- (i)  $K$  là số có 5 chữ số;
- (ii)  $K$  là số nguyên tố;
- (iii) Mỗi chữ số của  $K$  cũng là những số nguyên tố;
- (iv) *Tổng các chữ số của  $K$  là một số thuận nghịch hai chữ số;*
- (v) *Tích các chữ số của  $K$  là một số thuận nghịch ba chữ số.*

**40.** Hãy viết chương trình liệt kê tất cả các số tự nhiên  $K[]$  thỏa mãn đồng thời những điều kiện dưới đây:

- (i)  $K$  là số có 5 chữ số;
- (ii)  $K$  là số nguyên tố;
- (iii) Mỗi chữ số của  $K$  cũng là các số nguyên tố;
- (iv) *Tổng các chữ số của  $K$  là một số chia hết cho  $P$  ( $P$  được nhập từ bàn phím);*
- (v) *Tích các chữ số của  $K$  là một số chia hết cho  $Q$  ( $Q$  được nhập từ bàn phím);*
- (vi) *Các chữ số của  $K$  không chứa số  $R$  (được nhập từ bàn phím).*

**41.** Hãy viết chương trình liệt kê tất cả các số tự nhiên  $K[]$  thỏa mãn đồng thời những điều kiện dưới đây:

- (i)  $K$  là số có 5, 7 hoặc 9 chữ số;
- (ii)  $K$  là số thuận nghịch;
- (iii) *Tổng các chữ số của  $K$  là một số chia hết cho  $P$  ( $P$  được nhập từ bàn phím);*

- (iv) Tích các chữ số của  $K$  là một số chia hết cho  $Q$  ( $Q$  được nhập từ bàn phím);
- (v) Các chữ số của  $K$  không chứa số  $R$  (được nhập từ bàn phím).

**42.** Cho dãy  $A[]$  gồm  $N$  số tự nhiên khác nhau và số tự nhiên  $K$ . Hãy viết chương trình liệt kê tất cả các dãy con của dãy số  $A[]$  sao cho tổng các phần tử trong dãy con đó đúng bằng  $K$ . Dữ liệu vào cho bởi file dayso.in theo khuôn dạng sau:

- Dòng đầu tiên ghi lại số tự nhiên  $N$  và  $K$ ; hai số được viết cách nhau bởi một vài khoảng trống;
- Dòng kế tiếp ghi lại  $N$  số của dãy số  $A[]$ , hai số được viết cách nhau một vài khoảng trống.

Các dãy con thoả mãn điều kiện tìm được ghi lại trong file ketqua.out theo khuôn dạng sau:

- Dòng đầu tiên ghi lại số các dãy con có tổng các phần tử đúng bằng  $K$  tìm được;
- Những dòng kế tiếp mỗi dòng ghi lại một dãy con. Hai phần tử khác nhau của dãy con được viết cách nhau bởi một vài khoảng trống.

Ví dụ dưới đây sẽ minh hoạ cho file dayso.in và ketqua.out của bài toán.

**Dayso.in**

```
5      50
5      10      15      20      25
```

**Ketqua.out**

```
3
10      15      25
5      20      25
5      10      15      20
```

**43.** Cho dãy  $A[]$  gồm  $N$  số tự nhiên khác nhau, số tự nhiên  $K$  và số tự nhiên  $B$ . Hãy viết chương trình liệt kê tất cả các dãy con  $K$  phần tử của dãy số  $A[]$  sao cho tổng các phần tử trong dãy con đó đúng bằng  $B$ . Dữ liệu vào cho bởi file dayso.in theo khuôn dạng sau:

- Dòng đầu tiên ghi lại ba số tự nhiên  $N, K, B$ ; Ba số được viết cách nhau bởi một vài khoảng trống;
- Dòng kế tiếp ghi lại  $N$  số của dãy số  $A[]$ , hai số được viết cách nhau một vài khoảng trống.

Các dãy con  $K$  phần tử thoả mãn điều kiện tìm được ghi lại trong file ketqua.out theo khuôn dạng sau:

- Dòng đầu tiên ghi lại số các dãy con  $K$  phần tử có tổng các phần tử đúng bằng  $B$  tìm được;
- Những dòng kế tiếp mỗi dòng ghi lại một dãy con. Hai phần tử khác nhau của dãy con được viết cách nhau bởi một vài khoảng trống.

Ví dụ dưới đây sẽ minh hoạ cho file dayso.in và ketqua.out của bài toán.

**Dayso.in**

```
5      3      50
5      10     15     20     25
```

**Ketqua.out**

```
2
10     15     25
5      20     25
```



**44.** Cho dãy gồm  $N$  số nguyên phân biệt  $A[] = \{a_1, a_2, \dots, a_N\}$  và số tự nhiên  $K$  ( $K \leq N \leq 100$ ). Hãy viết chương trình liệt kê tất cả các dãy con  $K$  phần tử tăng dần của dãy số  $A[]$ . Dữ liệu vào cho bởi file dayso.in theo khuôn dạng sau:

- i. Dòng đầu tiên ghi lại hai số tự nhiên  $N, K$ . Hai số được viết cách nhau một vài khoảng trống;
- ii. Những dòng kế tiếp ghi lại  $N$  số nguyên của dãy số  $A[]$ , hai số khác nhau được viết cách nhau một vài khoảng trống.

Các dãy con  $K$  phần tử tăng dần của dãy số  $A[]$  tìm được ghi lại trong file ketqua.out theo khuôn dạng:

- Dòng đầu tiên ghi lại số tự nhiên  $M$  là số các dãy con  $K$  phần tử tăng dần của dãy số  $A[]$  tìm được;
- $M$  dòng kế tiếp, mỗi dòng ghi lại một dãy con. Hai phần tử khác nhau của dãy con được viết cách nhau bởi một vài khoảng trống.

Ví dụ với file dayso.in dưới đây sẽ cho ta file ketqua.out tương ứng.

**dayso.in**

```
5      3
2      5      15      10      20
```

**ketqua.out**

```
7
2      5      15
2      5      10
2      5      20
2      15     20
2      10     20
5      15     20
5      10     20
```