Cơ sở lập trình
Bài 7. Mảng một chiều và chuỗi ký tự (6 tiết)

Biên soạn TS. Trần Minh Thái
Giảng viên: Lê Thị Minh Nguyện
Email: nguyenltm@huflit.edu.vn

Mục tiêu

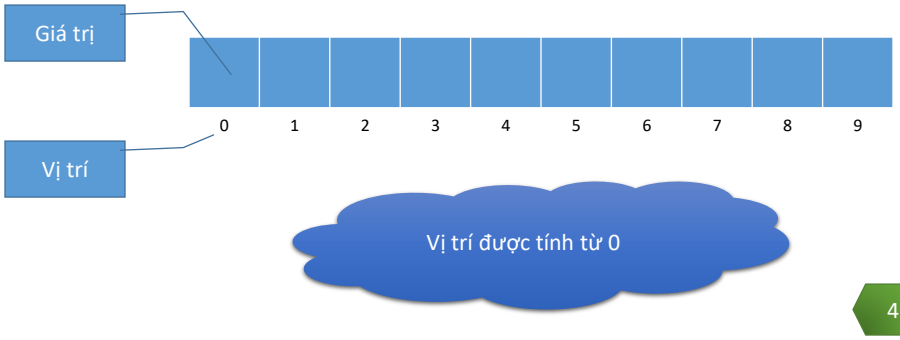
1. Trình bày khái niệm về kiểu dữ liệu mảng một chiều
2. Cách khai báo và khởi gán giá trị
3. Các thao tác xử lý trên mảng một chiều
4. Xử lý chuỗi ký tự

Nhu cầu

3

Khái niệm

- Được cấp phát bộ nhớ liên tục
- Tập hợp các biến có cùng kiểu dữ liệu và cùng tên. Truy xuất các biến thành phần thông qua chỉ mục (vị trí)



4

Khai báo

<Kiểu dữ liệu> <Tên mảng> [<Số phần tử tối đa của mảng>] ;

- ***int a[100]; //Khai bao mang so nguyen a gom 100 phan tu***
- ***float b[50]; //Khai bao mang so thuc b gom 50 phan tu***
- ***char str[30]; //Khai bao mang ky tu str gom 30 ky tu***

Nhằm thuận tiện cho việc viết chương trình, ta nên định nghĩa hằng số MAX ở đầu chương trình – là kích thước tối đa của mảng - như sau:

```
#define MAX 100
int main()
{
        int a[MAX], b[MAX];
        //Các lệnh
        return 0;
}
```

5

Khai báo và khởi gán

Gán từng phần tử

int a[5] = {3, 6, 8, 1, 12};

Giá trị	3	6	8	1	12
Vị trí	*0*	*1*	*2*	*3*	*4*

Gán toàn bộ phần tử có cùng giá trị

int a[8] = {3};

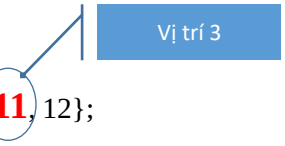
Giá trị	3	3	3	3	3	3	3	3
Vị trí	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*

6

Truy xuất giá trị

TênMảng [vị trí cần truy xuất]

```
int main()
{
        int a[5] = {3, 6, 8, 11, 12};
        printf( “Gia tri mang tai vi tri 3 = %d”, a[3]);
        return 0;
}
```



Kết quả: Gia tri mang tai vi tri 3 = 11

7

Các thao tác trên mảng

1. Nhập/ khởi gán
2. Xuất (liệt kê/ lọc)
3. Tìm kiếm
4. Đếm
5. Sắp xếp
6. Kiểm tra mảng thỏa điều kiện cho trước
7. Tách/ ghép mảng
8. Chèn / xóa

8

Nhập mảng

Giá trị	a[0]	a[1]	a[2]	a[3]	a[4]	…	…	a[n-2]	a[n-1]
Vị trí	0	1	2	3	4	…	…	n-2	n-1

```
void NhapMang (int a[], int n)
{
    for (int i = 0; i < n; i ++)
    {
      printf("Nhap phan tu thu %d: ", i);
      scanf("%d", &a[i]);
    }
}
```

9

Xuất mảng

Giá trị	a[0]	a[1]	a[2]	a[3]	a[4]	…	…	a[n-2]	a[n-1]
Vị trí	0	1	2	3	4	…	…	n-2	n-1

```
void XuatMang (int a[], int n)
{
    for (int i = 0; i < n; i ++)
    {
      printf("%d\t", a[i]);
    }
}
```

10

Ví dụ 5.1

Viết chương trình nhập vào mảng một chiều số nguyên, kích thước n. Xuất ra màn hình các giá trị của mảng vừa nhập.

Chương trình gồm có các yêu cầu:

1. ***Nhập mảng một chiều có kích thước n***
2. ***Xuất các giá trị của mảng***

11

```c
#include<stdio.h>
#define MAX 100

void NhapMang(int a[], int n);
void XuatMang(int a[], int n);

void NhapMang(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("* Nhap vao phan tu tai vi tri %d: ", i);
        scanf("%d", &a[i]);
    }
}
void XuatMang(int a[], int n)
{
    for(int i=0; i<=n-1;i++)
    {
        printf("%d ",a[i]);
    }
}
```

```c
int main()
{
    int a[MAX], n;
    printf("So luong phan tu cua mang = ");
    scanf("%d", &n);
    NhapMang(a, n);
    printf("Cac gia tri trong mang a:\n");
    XuatMang(a, n);

    return 0;
}
```

12

Bài tập

Viết chương trình:

1. Nhập vào mảng một chiều số thực, kích thước n sao cho $0 < n \leq MAX$ (*MAX là kích thước tối đa của mảng*)
2. Xuất ra màn hình các giá trị của mảng vừa nhập.

13

Phát sinh giá trị ngẫu nhiên

- **Thư viện**: <stdlib.h> và <time.h>
- **Khởi tạo**: *Chỉ gọi 1 lần trong chương trình trước khi sinh ngẫu nhiên*

  srand((unsigned int)time(NULL));
- **Hàm sinh giá trị ngẫu nhiên**:

  rand()%n → trả về giá trị nguyên từ 0 đến n-1
- ***Tạo giá trị ngẫu nhiên cho kq từ 0 đến 40***

```
srand((unsigned int)time(NULL));
int kq = rand()%41;
```

14

Bài tập

5.1. Viết chương trình tạo ngẫu nhiên các phần tử cho mảng một chiều số nguyên (*kích thước n*) sao cho giá trị ∈**[1.. n]**

5.2. Viết hàm tạo ngẫu nhiên các phần tử cho mảng một chiều số nguyên (*kích thước n*) sao cho giá trị ∈**[-n..n]**

5.3. Viết hàm tạo ngẫu nhiên các phần tử cho mảng một chiều số nguyên (*kích thước n*) sao cho giá trị **> 0** và **có thứ tự tăng dần**

15

Liệt kê các phần tử thỏa đk cho trước

Mẫu 1:

```
void LietKeXXX(int a[], int n)
{
  for (int i = 0; i<n; i++)
        if (a[i] thỏa điều kiện)
              Xuất a[i];
}
```

Mẫu 2:

```
void LietKeXXX(int a[], int n, int x)
{
    for (int i = 0; i<n; i++)
        if (a[i] thỏa điều kiện so với x)
              Xuất a[i];
}
```

16

Ví dụ: Liệt kê các phần tử có giá trị lẻ trong mảng

```
void XuatLe(int a[], int n)
{
   for (int i = 0; i<n; i++)
      if (a[i] %2 !=0)
            printf("%d\t", a[i]);
}
```

Ví dụ: Liệt kê các phần tử có giá trị lớn hơn x trong mảng

```
void XuatLonHonX(int a[], int n, int x)
{
   for (int i = 0; i<n; i++)
      if (a[i] > x)
            printf("%d\t", a[i]);
}
```

Ví dụ 5.2

Viết chương trình nhập vào mảng một chiều số nguyên gồm n phần tử. In ra màn hình những phần tử có giá trị lẻ.

```c
#include<stdio.h>
#define MAX 100

void NhapMang(int a[], int n);
void XuatMang(int a[], int n);
void XuatLe(int a[], int n);

void NhapMang(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("* Nhap vao phan tu tai vi tri %d: ", i);
        scanf("%d", &a[i]);
    }
}
void XuatMang(int a[], int n)
{
    for(int i=0; i<=n-1;i++)
    {
        printf("%d ",a[i]);
    }
}
```

```c
void XuatLe(int a[], int n)
{
    for(int i=0; i<=n-1;i++)
    {
        if(a[i]%2!=0)
            printf("%d ",a[i]);
    }
}
int main()
{
    int a[MAX], n;
    printf("So luong phan tu cua mang = ");
    scanf("%d", &n);
    NhapMang(a, n);
    printf("Cac gia tri trong mang a:\n");
    XuatMang(a, n);
    printf("\nCac gia tri le trong mang a:\n");
    XuatLe(a, n);
    return 0;
}
```

Ví dụ 5.3

Viết chương trình nhập vào mảng một chiều số nguyên, kích thước n. In ra các phần tử có giá trị lớn hơn x có trong mảng

```c
#include<stdio.h>
#define MAX 100

void NhapMang(int a[], int n);
void XuatMang(int a[], int n);
void XuatLonHonX(int a[], int n, int x);

void NhapMang(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("* Nhap vao phan tu tai vi tri %d: ", i);
        scanf("%d", &a[i]);
    }
}
void XuatMang(int a[], int n)
{
    for(int i=0; i<=n-1;i++)
    {
        printf("%d ",a[i]);
    }
}
```

```c
void XuatLonHonX(int a[], int n, int x)
{
    for(int i=0; i<=n-1;i++)
    {
        if(a[i]>x)
            printf("%d ",a[i]);
    }
}
```

```c
int main()
{
    int a[MAX], n, x;
    printf("So luong phan tu cua mang = ");
    scanf("%d", &n);
    NhapMang(a, n);
    printf("Cac gia tri trong mang a:\n");
    XuatMang(a, n);
    printf("\nGia tri x can so sanh = ");
    scanf("%d", &x);
    printf("\nCac gia tri lon hon x = %d:\n", x);
    XuatLonHonX(a, n, x);
    return 0;
}
```

21

Bài tập

Viết các hàm sau:

5.4. Xuất các phần tử là **bội số của 5** trong mảng

5.5. Xuất các phần tử là **số nguyên tố** trong mảng

5.6. Xuất các phần tử tại các **vị trí lẻ** trong mảng

22

Đếm số lượng phần tử

Mẫu 1:

```
int DemXXX(int a[], int n)
{
    int d = 0;
    for (int i = 0; i<n; i++)
        if (a[i] thỏa điều kiện)
            d++;
    return d;
}
```

23

Ví dụ: Đếm các phần tử có giá trị là số nguyên tố

```
int DemUS(int k)
{
    int d = 0;
    for (int i = 1; i <= k; i++)
        if (k % i == 0)
            d++;
    return d;
}
int LaSNT(int k)
 {
    if(DemUS(k)==2)
        return 1;
    return 0;
}
```

```
int DemSNT(int a[], int n)
{
  int d = 0;
  for (int i = 0; i<n; i++)
  {
    if (LaSNT(a[i]) ==1)
    {
        d++;
    }
  }
  return d;
}
```

24

Ví dụ 5.4: Chương trình nhập vào mảng một chiều số nguyên, kích thước n. Đếm số lượng các phần tử là số nguyên tố có trong mảng

```c
#include<stdio.h>
#define MAX 100

void NhapMang(int a[], int n);
void XuatMang(int a[], int n);
int LaSNT(int k);
int DemSNT(int a[], int n);

void NhapMang(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("* Nhap vao phan tu tai vi tri %d: ", i);
        scanf("%d", &a[i]);
    }
}
void XuatMang(int a[], int n)
{
    for(int i=0; i<=n-1;i++)
    {
        printf("%d ",a[i]);
    }
}
```

25

```c
int LaSNT(int k)
{
    int d = 0;
    for (int i = 1; i <= k; i++)
    {
        if (k % i == 0)
            d++;
    }
    if(d==2)
        return 1;
    return 0;
}
int DemSNT(int a[], int n)
{
    int d = 0;
    for (int i = 0; i<n; i++)
    {
        if (LaSNT(a[i]) == 1)
            d++;
    }
    return d;
}
```

26

```
46  int main()
47  {
48      int a[MAX], n, kq;
49      printf("So luong phan tu cua mang = ");
50      scanf("%d", &n);
51      NhapMang(a, n);
52      printf("Cac gia tri trong mang a:\n");
53      XuatMang(a, n);
54
55      kq = DemSNT(a, n);
56      if(kq==0)
57          printf("\n>> Khong co so nguyen to trong mang");
58      else
59          printf("\n>> So luong so nguyen to = %d", kq);
60
61      return 0;
62  }
```

27

Đếm số lượng phần tử

Mẫu 2:

```
int DemXXX(int a[], int n, int x)
{
    int d = 0;
    for (int i = 0; i<n; i++)
        if (a[i] thỏa điều kiện so với x)
            d++;
    return d;
}
```

28

Ví dụ: Đếm các phần tử có giá trị nhỏ hơn x có trong mảng

```
int DemNhoHonX(int a[], int n, int x)
{
    int d = 0;
    for (int i = 0; i<n; i++)
        if (a[i] < x)
            d++;
   return d;
}
```

29

Ví dụ 5.5: Chương trình nhập vào mảng một chiều số nguyên, kích thước n. Đếm số lượng các phần tử có giá trị nhỏ hơn x có trong mảng

```
1  #include<stdio.h>
2  #define MAX 100
3
4  void NhapMang(int a[], int n);
5  void XuatMang(int a[], int n);
6  int DemNhoHonX(int a[], int n, int x);
7
8  void NhapMang(int a[], int n)
9  {
10     for (int i = 0; i < n; i++)
11     {
12         printf("* Nhap vao phan tu tai vi tri %d: ", i);
13         scanf("%d", &a[i]);
14     }
15 }
16 void XuatMang(int a[], int n)
17 {
18     for(int i=0; i<=n-1;i++)
19     {
20         printf("%d ",a[i]);
21     }
22 }
```

30

```c
int DemNhoHonX(int a[], int n, int x)
{
    int d = 0;
    for (int i = 0; i<n; i++)
    {
        if (a[i] < x)
            d++;
    }
    return d;
}
```

```c
int main()
{
    int a[MAX], n, kq, x;
    printf("So luong phan tu cua mang = ");
    scanf("%d", &n);
    NhapMang(a, n);
    printf("Cac gia tri trong mang a:\n");
    XuatMang(a, n);

    printf("\nNhap gia tri x = ");
    scanf("%d", &x);
    kq = DemNhoHonX(a, n, x);
    if(kq==0)
        printf("\n>> Khong co phan tu nho hon %d", x);
    else
        printf("\n>> So luong phan tu < %d = %d", x, kq);

    return 0;
}
```

31

Bài tập

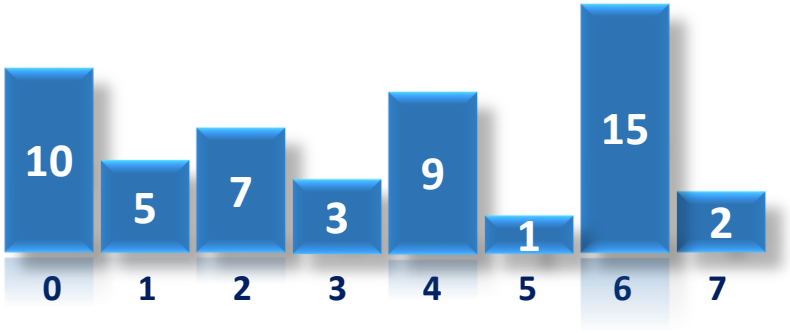
Cho mảng một chiều số nguyên a, kích thước n, viết hàm:

1. Đếm số lượng các **phần tử lẻ**
2. Đếm số lượng các phần tử là **ước số của x**

32

Tìm vị trí phần tử nhỏ nhất

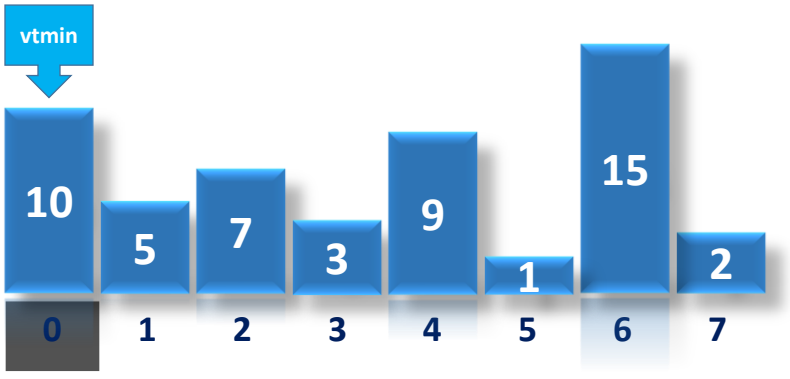
Cần tìm vị trí phần tử nhỏ nhất trong dãy số sau?



33

Tìm vị trí phần tử nhỏ nhất

B1: Giả sử vị trí phần tử nhỏ nhất là 0 (vtmin) – giá trị 10



34

Tìm vị trí phần tử nhỏ nhất

B2: So sánh giá trị tại vtmin với tất cả giá trị tại các vị trí còn lại (từ 1 đến 7), nếu có phần tử nào < phần tử tại vtmin thì cập nhật lại vtmin

35

Tìm vị trí phần tử nhỏ nhất

B2: So sánh giá trị tại vtmin với tất cả giá trị tại các vị trí còn lại (từ 1 đến 7), nếu có phần tử nào < phần tử tại vtmin thì cập nhật lại vtmin

36

Tìm vị trí phần tử nhỏ nhất

B2: So sánh giá trị tại vtmin với tất cả giá trị tại các vị trí còn lại (từ 1 đến 7), nếu có phần tử nào < phần tử tại vtmin thì cập nhật lại vtmin

37

Tìm vị trí phần tử nhỏ nhất

B2: So sánh giá trị tại vtmin với tất cả giá trị tại các vị trí còn lại (từ 1 đến 7), nếu có phần tử nào < phần tử tại vtmin thì cập nhật lại vtmin

38

Tìm vị trí phần tử nhỏ nhất

B2: So sánh ... với tất cả giá trị tại các vị trí còn lại (từ 1 đến 7), nếu ... tử tại vtmin thì cập nhật lại vtmin

39

Tìm vị trí phần tử nhỏ nhất

B2: So sánh ... với tất cả giá trị tại các vị trí còn lại (từ 1 đến 7), nếu ... tử tại vtmin thì cập nhật lại vtmin

40

Tìm vị trí phần tử nhỏ nhất

B2: So sánh giá trị tại vtmin với tất cả giá trị tại các vị trí còn lại (từ 1 đến 7), nếu có phần tử nào < phần tử tại vtmin thì cập nhật lại vtmin

41

Hàm tìm vị trí phần tử có giá trị nhỏ nhất

```
int TimVTMin(int a[], int n)
{
    int vtmin = 0;
    for (int i = 1; i < n; i++)
    {
      if (a[i] < a[vtmin])
            vtmin = i;
    }
    return vtmin;
}
```

42

Bài tập

5.7. Viết hàm tìm vị trí phần tử có giá trị lớn nhất

5.8. Viết hàm tìm vị trí phần tử có giá trị dương nhỏ nhất

5.9. Viết hàm tìm vị trí phần tử có giá trị âm lớn nhất

Làm tại lớp bài tập 5.7 với 5.9

43

Tìm phần tử x

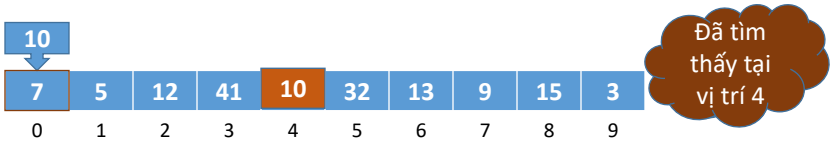
- **Ý tưởng**

  Lần lượt so sánh **x** với phần tử thứ nhất, thứ hai, ... của mảng **a** cho đến khi gặp được phần tử cần tìm, hoặc đã tìm hết mảng mà không thấy **x**

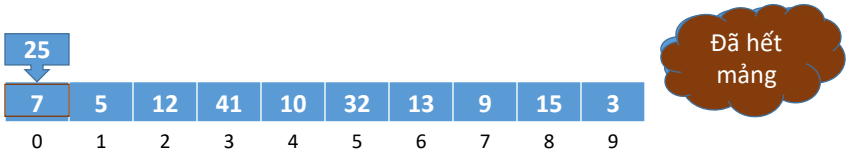
44

Tìm phần tử x

- Tìm x = 10



- Tìm x = 25



45

```
int TimVTX(int a[], int n, int x)
{
    for (int i = 0; i < n; i++)
    {
        if (a[i] == x)
            return i;
    }
    return -1;
}
```

Tìm phần tử x: Nếu tìm thấy trả về vị trí; ngược lại trả về -1

Bài tập

1. Viết hàm tìm vị trí phần tử có giá trị x (*nếu có*). Nếu mảng có nhiều giá trị x thì trả về vị trí x xuất hiện cuối cùng trong mảng

2. Viết hàm tìm và lưu tất cả các vị trí của phần tử có giá trị x (*nếu có*)

47

Kiểm tra xem mảng có thỏa điều kiện

- **TH1**: **kiểm tra tồn tại** một phần tử trong mảng thỏa điều kiện nào đó cho trước → tìm phần tử thỏa điều kiện để kết luận
- **TH2**: **kiểm tra tất cả** các phần tử thỏa điều kiện nào đó cho trước → tìm phần tử không thỏa điều kiện để kết luận mảng không thỏa điều kiện

48

Các trường hợp kiểm tra mảng

Mẫu TH1:

```
int KiemTraTonTaiXXX(int a[], int n)
{
        for (int i = 0; i<n; i++)
                if (a[i] thỏa điều kiện)
                        return 1;
        return 0;
}
```

Mẫu TH2:

```
int  KiemTraXXX(int a[], int n)
{
        for (int i = 0; i<n; i++)
                if (a[i] không thỏa điều kiện)
                        return 0;
         return 1;
 }
```

49

Ví dụ: Kiểm tra xem mảng có tồn tại số lẻ không?

```
int KiemTraTonTaiLe(int a[], int n)
{
    for (int i = 0; i < n; i++)
    {
        if (a[i] % 2 != 0)
            return 1;
    }
    return 0;
}
```

50

Ví dụ: Kiểm tra xem mảng có chứa toàn giá trị âm?

```
int KiemTraToanAm(int a[], int n)
{
    for (int i  = 0; i<n; i++)
    {
        if (a[i] >= 0)
            return 0;
    }
    return 1;
}
```

51

Bài tập

5.11. Viết hàm kiểm tra xem mảng có tồn tại số nguyên tố không?

5.12. Viết hàm kiểm tra xem mảng có âm dương xen kẻ không?

5.13. Viết hàm kiểm tra xem mảng có chẵn lẻ xen kẻ không?

5.14. Viết hàm kiểm tra xem mảng có thứ tự tăng dần không?

52

Tính tổng, giá trị trung bình thỏa điều kiện

Mẫu tính tổng:

```
int TongXXX(int a[], int n)
{
    int s = 0;
    for (int i = 0; i<n; i++)
    {
        if (a[i] thỏa điều kiện)
            s += a[i];
    }
    return s;
}
```

53

Tính tổng, giá trị trung bình thỏa điều kiện

Mẫu tính trung bình:

```
float  TrungBinhXXX(int a[], int n)
{
    int s = 0;
    int d = 0;
    for (int i = 0; i<n; i++)
    {
        if (a[i] thỏa điều kiện)
        {
            s += giatri;
            d ++;
        }
    }
    if (d==0)
        return 0;
    return (float) s / d;
}
```

54

Ví dụ: Tính tổng các phần tử có giá trị lẻ trong mảng

```
int TongLe(int a[], int n)
{
        int s = 0;
        for (int i = 0; i<n; i++)
        {
           if (a[i] %2!=0)
              s += a[i];
        }
        return s;
}
```

55

Ví dụ: Tính giá trị trung bình các phần tử có giá trị âm

```
float TrungBinhAm(int a[], int n)
{
        long s = 0;
        int d = 0;
        for (int i  = 0; i<n; i++)
        {
           if (a[i] < 0)
           {
              s += a[i];
              d++;
           }
        }
        if (d == 0)
           return 0;
        return (float)s / d;
}
```

56

Bài tập

5.15. Viết hàm tính tổng các phần tử là số nguyên tố

5.16. Viết hàm tính giá trị trung bình các phần tử có trong mảng

5.17. Viết hàm tính giá trị trung bình các phần tử có giá trị nhỏ hơn x

57

Sắp xếp mảng

```
void SapTang(int a[], int n)
{
	for (int i = 0; i < n-1; i ++)
	 {
		for(int j = i+1; j < n; j ++)
			if (a[i] > a[j])
				HoanVi(&a[i], &a[j]);
	 }
}
void HoanVi(int *a, int *b)
{
	int tam = *a;
	*a = *b;
	*b = tam;
}
```

58

Bài tập

5.18. Viết hàm sắp xếp mảng theo thứ tự giảm dần

5.19. Viết hàm sắp xếp các phần tử lẻ của mảng theo thứ tự tăng dần

59

Chèn thêm phần tử vào mảng

- Cho mảng sau:

12	5	7	9	21	38
0	*1*	*2*	*3*	*4*	*5*

- Hãy trình bày từng bước chèn 111 vào vị trí 3 của mảng mà không dùng mảng phụ

111

12	5	7	9	21	38	
0	*1*	*2*	*3*	*4*	*5*	

60

Chèn thêm phần tử vào mảng

- Hãy viết hàm chèn phần tử có giá trị x vào vị trí k cho trước trong mảng a kích thước n theo mẫu sau:

void ChenX(int a[], int &n, int x, int k);

61

Bài tập

5.20. Viết hàm chèn x vào sau phần tử có giá trị nhỏ nhất (*giả sử mảng không có giá trị trùng nhau*)

5.21. Viết hàm chèn x vào sau phần tử có giá trị y xuất hiện đầu tiên trong mảng. Nếu không có y thì chèn vào đầu mảng

5.22. Giả sử có mảng a đã được sắp theo thứ tự tăng dần, hãy viết hàm chèn x vào mảng a sao cho mảng a vẫn giữ nguyên thứ tự tăng dần mà không cần sắp xếp lại mảng

62

Xóa phần tử khỏi mảng

- Cho mảng sau:

12	5	7	9	21	38
0	*1*	*2*	*3*	*4*	*5*

- Hãy trình bày từng bước xóa phần tử tại vị trí 3 trong mảng mà không dùng mảng phụ

12	5	7	9	21	38
0	*1*	*2*	*3*	*4*	*5*

63

Xóa phần tử khỏi mảng

- Hãy viết hàm xóa phần tử tại vị trí k cho trước trong mảng a kích thước n theo mẫu sau:

void XoaTaiVTk(int a[], int &n, int k);

64

Bài tập

5.23. Viết hàm xóa phần tử có giá trị lớn nhất (*giả sử mảng không có giá trị trùng nhau*)

5.24. Giả sử mảng có nhiều giá trị x, hãy viết hàm xóa những phần tử có giá trị trùng với x sao cho chỉ giữ lại 1 phần tử

65

Bài tập

5.25. Cho mảng một chiều số nguyên a. Hãy tạo 2 mảng b và c sao cho mảng b chứa những số lẻ và mảng c chứa những phần tử còn lại

5.26. Cho viết chương trình nhập vào 2 mảng một chiều số nguyên a và b.

1. Sắp xếp a và b theo thứ tự tăng dần
2. Trộn 2 mảng a và b thành mảng c sao cho c có thứ tự tăng dần mà không cần sắp xếp

66

Q&A

CHUỖI KÝ TỰ

Chuỗi ký tự

- Chuỗi ký tự là trường hợp đặc biệt của mảng 1 chiều, là một dãy các phần tử, mỗi phần tử có kiểu ký tự
- Khai báo:

char < Tên chuỗi > [< Số ký tự tối đa>] ;

Ví dụ: *char chuoi[25];*

Ý nghĩa khai báo **1 mảng kiểu ký tự tên là chuoi** có 25 phần tử (như vậy tối đa ta có thể nhập 24 ký tự vì **phần tử thứ 25 đã chứa ký tự kết thúc chuỗi '\0'**)

- Lưu ý: Chuỗi ký tự được kết thúc bằng ký tự '\0'. Do đó khi khai báo độ dài của chuỗi luôn luôn khai báo dư 1 phần tử để chứa ký tự '\0'.

69

Khái niệm

- Ví dụ: Chuỗi *"NGUYEN VAN A"* được lưu

'N'	'G'	'U'	'Y'	'E'	'N'	' '	'V'	'A'	'N'	' '	'A'	'\0'
0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*

70

Nhập chuỗi

- gets(biến chuỗi);
- Ví dụ:

  char str[50];

  gets(str);

- Xóa bộ nhớ đệm: fflush(stdin);

!!! Giữa nhập số và chuỗi phải dùng lệnh xóa bộ nhớ đệm

71

Ví dụ

- Chương trình nhập vào điểm trung bình và họ tên của 1 sinh viên

72

Các hàm thư viện <string.h>

- Tính độ dài của chuỗi s

  int strlen(char s[]);
- Sao chép nội dung chuỗi nguồn vào chuỗi đích

  strcpy(char đích[], char nguồn[]);
- Chép n ký tự từ chuỗi nguồn sang chuỗi đích. Nếu chiều dài nguồn < n thì hàm sẽ điền khoảng trắng cho đủ n ký tự vào đích

  strncpy(char đích[], char nguồn[], int n);

  *** phải có: đích[n]='\0';

73

Các hàm thư viện <string.h>

- Nối chuỗi s2 vài chuỗi s1

  strcat(char s1[],char s2[]);
- Nối n ký tự đầu tiên của chuỗi s2 vào chuỗi s1

  strncat(char s1[],char s2[],int n);
- So sánh 2 chuỗi s1 và s2 theo nguyên tắc thứ tự từ điển. ***Phân biệt chữ hoa và thường.*** Trả về:

  0 : nếu s1 bằng s2.

  >0: nếu s1 lớn hơn s2.

  <0: nếu s1 nhỏ hơn s2.

  int strcmp(char s1[],char s2[]);

74

Các hàm thư viện <string.h>

- So sánh n ký tự đầu tiên của s1 và s2, giá trị trả về tương tự hàm strcmp()

  int strncmp(char s1[],char s2[], int n);

- So sánh chuỗi s1 và s2 nhưng không phân biệt hoa thường, giá trị trả về tương tự hàm strcmp()

  int stricmp(char s1[],char s2[]);

- So sánh n ký tự đầu tiên của s1 và s2 nhưng không phân biệt hoa thường, giá trị trả về tương tự hàm strcmp()

  int strnicmp(char s1[],char s2[], int n);

75

Các hàm thư viện <string.h>

- Tìm sự xuất hiện đầu tiên của ký tư c trong chuỗi s. Trả về:

  NULL: nếu không có

  Địa chỉ c: nếu tìm thấy

  char *strchr(char s[], char c);

- Tìm sự xuất hiện đầu tiên của chuỗi s2 trong chuỗi s1. Trả về:

  NULL: nếu không có

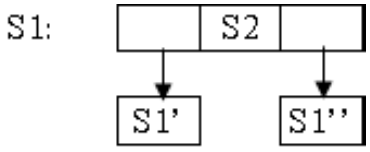
  Ngược lại: Địa chỉ bắt đầu chuỗi s2 trong s1

  char *strstr(char s1[], char s2[]);

76

Các hàm thư viện <string.h>

- Tách chuỗi:
  - Nếu s2 có xuất hiện trong s1: Tách chuỗi s1 thành hai chuỗi: Chuỗi đầu là những ký tự cho đến khi gặp chuỗi s2 đầu tiên, chuỗi sau là những ký tự còn lại của s1 sau khi đã bỏ đi chuỗi s2 xuất hiện trong s1.



  - Nếu s2 không xuất hiện trong s1 thì kết quả chuỗi tách vẫn là s1.

char *strtok(char s1[], char s2[]);

77

Bài tập

1. Viết hàm đếm số ký tự trắng trong chuỗi
2. Viết hàm kiểm tra xem chuỗi có đối xứng hay không?
3. Viết hàm xuất chuỗi đảo ngược các ký tự trong chuỗi

Ví dụ: nhập ABCDE,

Chuỗi đảo ngược là:EDCBA

4. Viết hàm tìm ký tự x có xuất hiện trong chuỗi không?

78

Q&A