



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

批量数据的遍历与访问

◆ 1、数组元素的遍历

6.1.3 一维数组的引用

- ▶ 从数组的内存形式来看，数组元素的下标是有序递增的，这个特点使得可以利用循环来批量处理数组元素。

6.1.3 一维数组的引用

- ▶ (1) 遍历数组元素
- ▶ 编写程序按指定要求访问数组元素，特别地，按顺序访问每个数组元素（遍历）。

6.1.3 一维数组的引用



【例6.1】

连续输入100个数，然后反序输出。

6.1.3 一维数组的引用

例6.1

```
1 #include<stdio.h>
2 int main()
3 {
4     int i, A[100]; //定义100个整型
5     for (i=0; i<100; i++) //连续输入100个数存储下来
6         scanf("%d",&A[i]);
7     for (i=100-1; i>=0; i--) //反序输出100个数
8         printf("%d ",A[i]);
9     return 0;
10 }
```

6.2.3 多维数组的引用



【例6.3】

给一个二维数组输入数据，并以行列形式输出。

6.2.3 多维数组的引用

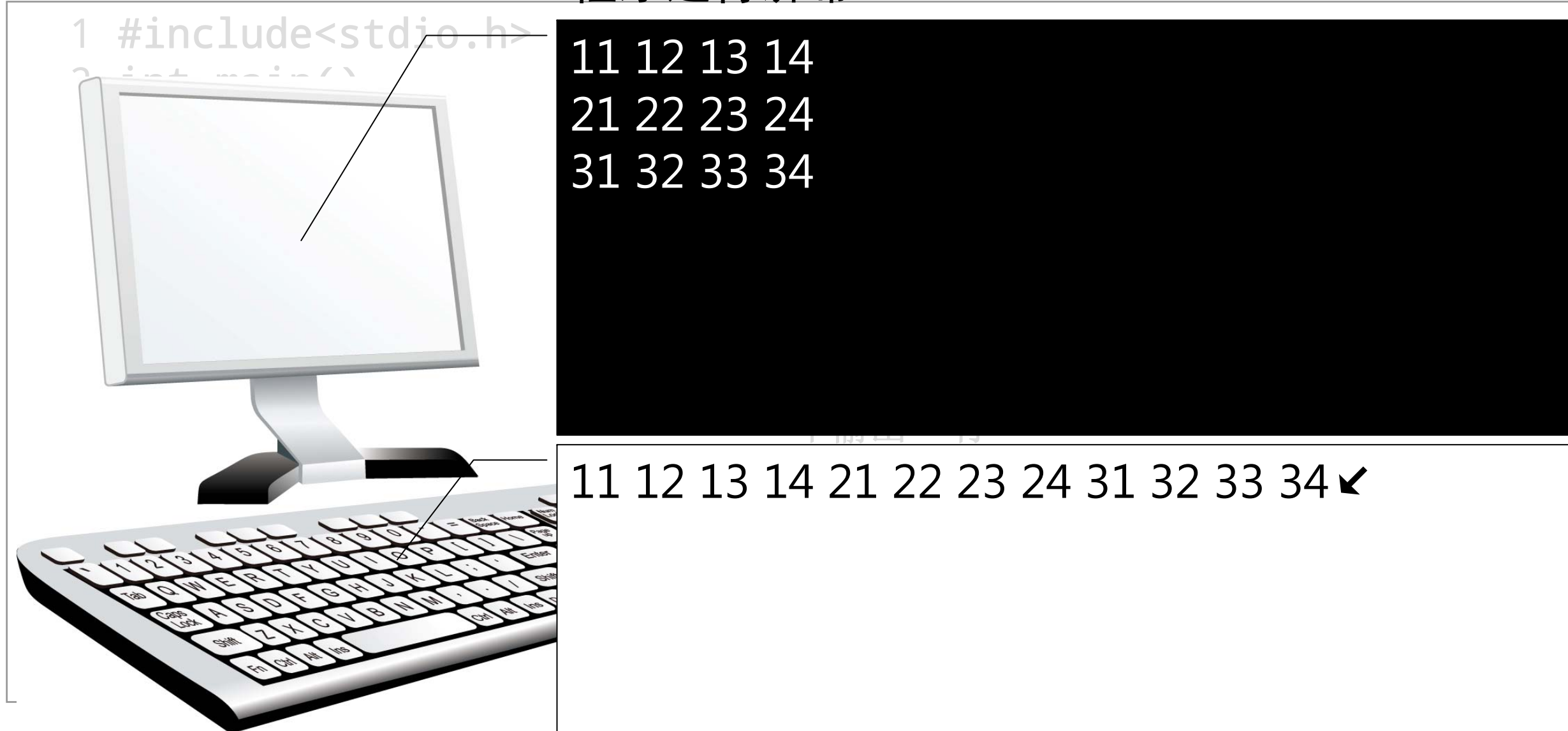
例6.3

```
1  #include<stdio.h>
2  int main()
3  {
4      int A[3][4],i,j; //二维数组下标应由两个独立的变量索引
5      for (i=0;i<3;i++) //双重循环遍历二维数组元素输入
6          for (j=0;j<4;j++)
7              scanf("%d",&A[i][j]);
8      for (i=0;i<3;i++) { //双重循环遍历二维数组元素输出
9          for (j=0;j<4;j++) //内循环输出一行
10             printf("%d ",A[i][j]);
11             printf("\n"); //每输出一行换行
12     }
13     return 0;
14 }
```


6.2.3 多维数组的引用

例6.3

程序运行屏幕

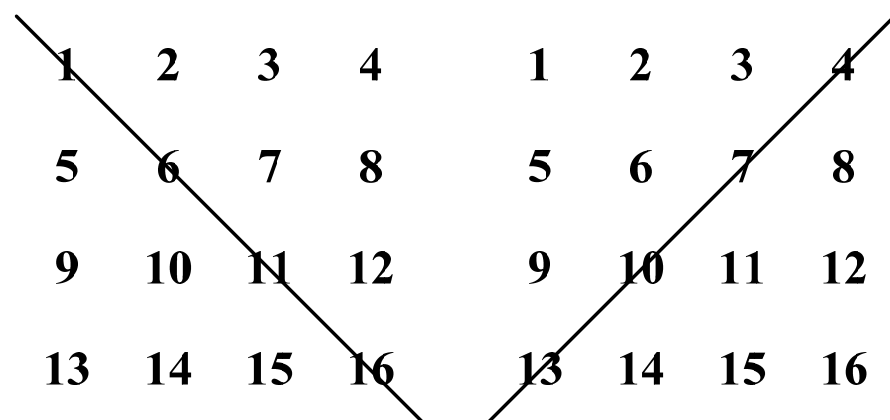


6.2.3 多维数组的引用



【数组元素遍历举例】

遍历二维数组对角线元素。



6.2.3 多维数组的引用

例6.51

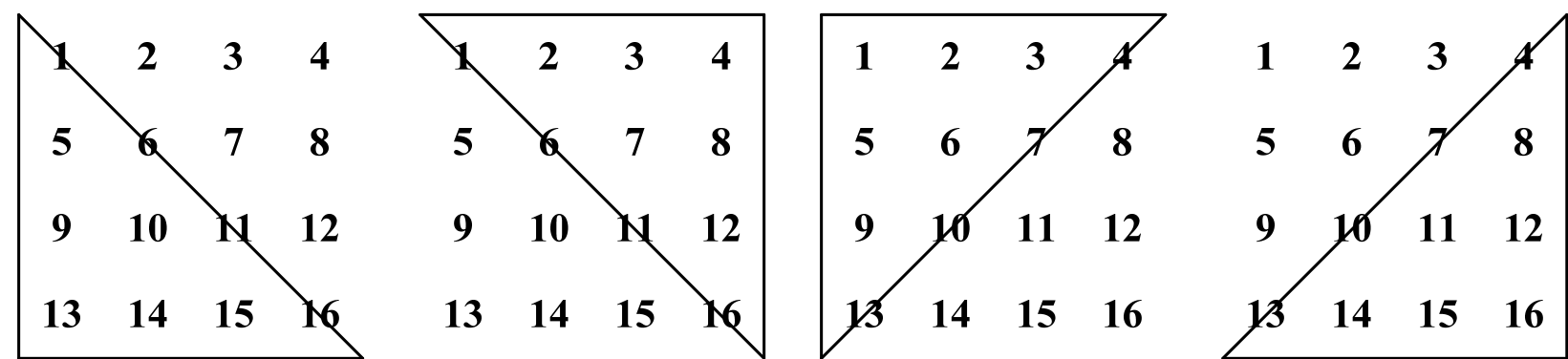
```
1 #include<stdio.h>
2 #define M 4
3 int main()
4 {
5     int A[M][M]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
6     int i , sum;
7     for (sum=0, i=0; i<M; i++)
8         sum = sum + A[i][i]; //对角线元素
9     printf("%d\n",sum); //输出累加和
10    for (sum=0, i=0; i<M; i++)
11        sum = sum + A[i][M-1-i]; //反对角线元素
12    printf("%d\n",sum); //输出累加和
13    return 0;
14 }
```

6.2.3 多维数组的引用



【数组元素遍历举例】

遍历二维数组四角元素。



6.2.3 多维数组的引用

例6.52

```
1  #include<stdio.h>
2  #define M 4
3  int main()
4  {
5      int A[M][M]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
6      int i ,j ,sum;
7      for (sum=0, i=0; i<M; i++) //左下角元素
8          for (j=0; j<=i; j++)
9              sum = sum + A[i][j]; //遍历指定条件元素
10     printf("%d\n",sum); //输出累加和
11     for (sum=0, i=0; i<M; i++) //右上角元素
12         for (j=i; j<M; j++)
13             sum = sum + A[i][j]; //遍历指定条件元素
14     printf("%d\n",sum); //输出累加和
15     for (sum=0, i=0; i<M; i++) //左上角元素
```

6.2.3 多维数组的引用

例6.52

```
16     for (j=0; j<M-i; j++)
17         sum = sum + A[i][j]; //遍历指定条件元素
18     printf("%d\n",sum); //输出累加和
19     for (sum=0, i=0; i<M; i++) //右下角元素
20         for (j=M-1-i; j<M; j++)
21             sum = sum + A[i][j]; //遍历指定条件元素
22     printf("%d\n",sum); //输出累加和
23     return 0;
24 }
```

6.2.3 多维数组的引用



【数组元素遍历举例】

遍历二维数组外围和内部区域元素。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

6.2.3 多维数组的引用

例6.53

```
1 #include<stdio.h>
2 #define M 4
3 int main()
4 {
5     int A[M][M]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
6     int i ,j ,sum;
7     for (sum=0, i=0; i<M; i++) //外围元素
8         for (j=0; j<M; j++)
9             if (i==0 || j==0 || i==M-1 || j==M-1)
10                 sum = sum + A[i][j]; //遍历指定条件元素
11     printf("%d\n",sum); //输出累加和
12     for (sum=0, i=0; i<M; i++) //内部元素
13         for (j=0; j<M; j++)
14             if (i>0 && i<M-1 && j>0 && j<M-1)
15                 sum = sum + A[i][j]; //遍历指定条件元素
```


6.2.3 多维数组的引用

例6.53

```
16    printf("%d\n",sum); //输出累加和
17    return 0;
18 }
```

6.1.3 一维数组的引用

- ▶ (2) 数组元素复制
- ▶ 通过元素逐个赋值，可以达到两个数组“赋值”的效果。

6.1.3 一维数组的引用



【例6.2】

将一维数组A“赋值”给B。

6.1.3 一维数组的引用

例6.2

```
1 #include<stdio.h>
2 int main()
3 {
4     int A[5]={1,2,3,4,5} , B[5] , i;
5     for (i=0; i<5; i++)
6         B[i] = A[i]; //元素一一复制
7     return 0;
8 }
```

6.2.3 多维数组的引用



【数组“赋值”举例】

复制二维数组A的元素到数组B中。

6.2.3 多维数组的引用

例6.54

```
1  #include<stdio.h>
2  #define M 3
3  #define N 3
4  int main()
5  {
6      int A[M][N] , B[M][N] , i , j;
7      for (i=0; i<M; i++)
8          for (j=0; j<N; j++)
9              scanf("%d",&A[i][j]); //一一输入A数组元素
10     for (i=0; i<M; i++)
11         for (j=0; j<N; j++)
12             B[i][j] = A[i][j]; //元素一一复制
13     for (i=0; i<M; i++) {
14         for (j=0; j<N; j++)
15             printf("%d ",B[i][j]); //一一输出B数组元素
```

6.2.3 多维数组的引用

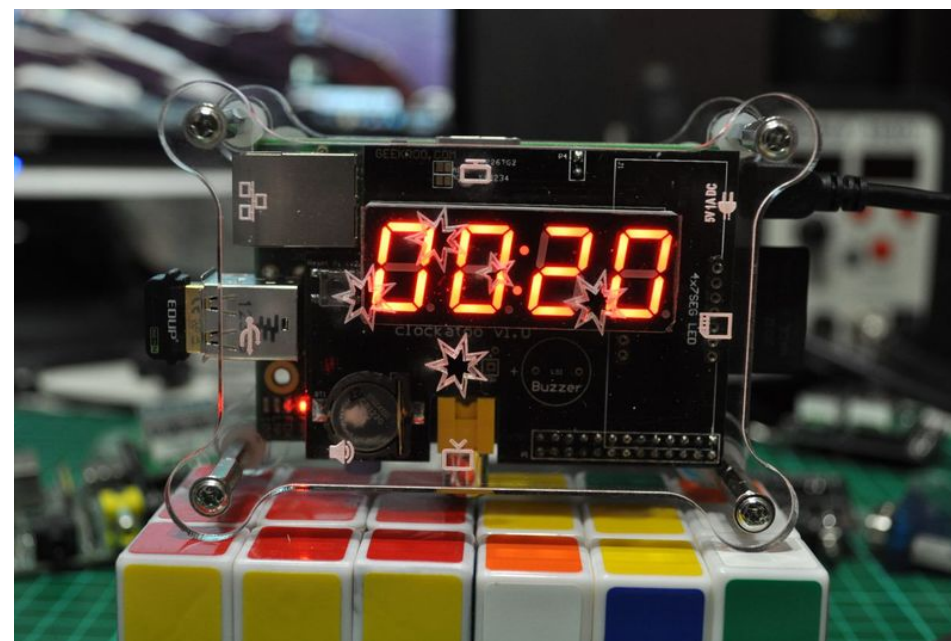
例6.54

```
16     printf("\n"); //每行换行
17 }
18 return 0;
19 }
```

6.2.3 多维数组的引用



Raspberry Pi（中文名为“树莓派”）是为学生计算机编程教育而设计，只有信用卡大小的卡片式电脑，如图所示。



6.2.3 多维数组的引用



【数组与枚举法】

树莓派的时钟有四个数字：两个小时两个分钟。例如，下面的时钟显示 9:30（注意前导零）。

09 : 30

所有数字如下显示。

0 9 0 9 0 9 0 9 0 9

6.2.3 多维数组的引用



【数组与枚举法】

输入：

数据有多组。每组1行，每行仅有1个整数 n ($0 \leq n \leq 30$)，表示高亮的显示段的数目。

6.2.3 多维数组的引用



【数组与枚举法】

输出：

对每个测试用例，输出“hh:mm”格式，表示高亮的显示段对应的时间（ $0 \leq hh < 24, 0 \leq mm < 60$ ），这样的时间可能有多个，输出最小的。如果没有对应的时间，输出“Impossible”

▶ 输入：

▶ 23

▶ 28

▶ 2

▶ 输出：

▶ 00:02

▶ Impossible

▶ Impossible

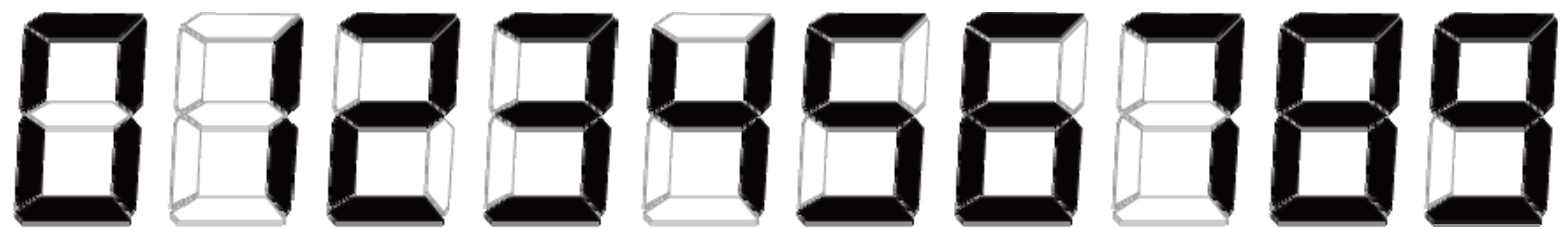
6.2.3 多维数组的引用



例题分析

分析：

根据



可以得到0-9的显示段数目，然后枚举hh和mm即可。其中约束条件：

$(0 \leq hh < 24, 0 \leq mm < 60)$

6.2.3 多维数组的引用

例6.66

```
1 #include <stdio.h>
2 int s[] = {6,2,5,5,4,5,6,3,7,6};
3 int main ()
4 {
5     int hh,mm,n,flag;
6     while (scanf("%d", &n)!=EOF) {
7         flag=0;
8         for (hh = 0; hh<24 && flag==0; hh++)
9             for (mm = 0; mm<60 && flag==0; mm++)
10                 if (s[hh%10]+s[hh/10]+s[mm%10]+s[mm/10]==n) {
11                     printf ("%02d:%02d\n", hh, mm);
12                     flag=1;
13                 }
14         if (flag==0)
15             printf ("Impossible\n");
```

6.2.3 多维数组的引用

例6.66

```
16     }  
17     return 0;  
18 }
```

6.2.3 多维数组的引用



如何调试有数组输入的程序

数组程序总是伴随大量数据的输入，例如：

```
for (i=0;i<100;i++)  
    scanf("%d",&A[i]);
```

每次运行程序，都需要输入100个输入，那么每次Debug，人岂不是要“累死”。

6.2.3 多维数组的引用



如何调试有数组输入的程序

现在，可以这样：

```
for (i=0;i<100;i++)  
    scanf("%d",&A[i]);
```

(1) 把那100个数先存放在一个记事本文件中，全选，复制到剪贴板。
每次程序运行时，粘贴而不是输入即可。

6.2.3 多维数组的引用



如何调试有数组输入的程序

现在，更建议这样：

```
freopen("in.txt","r",stdin); //增加这一行  
for (i=0;i<100;i++)  
    scanf("%d",&A[i]);
```

(2) freopen函数将标准输入stdin设置为一个文件，那么后面的scanf就不是从键盘输入，而是直接从文件in.txt输入了。

把那100个数先存放在一个记事本文件in.txt中，程序增加freopen函数一行。

CP 程序设计