C语言程序设计

计算机科学与技术学院

数组

- 一维数组
- 二维数组
- 字符数组
- 数组相关算法

内存中数据存储问题

- 如下几组数据在内存中如何存储?
  - 10个学生的学习成绩
  - 2行2列的矩阵
  - 一串字符

```
int cj1, cj2, cj3, cj4, cj5;
cj1=75; cj2=85; cj3=72; cj4=88;  cj5=90;

int a11,a12,a21,a22;
a11=1;a12=2;a21=3;a22=4;

char c1,c2,c3,c4;
c1='D'; c2='H'; c3='U'; c4='\n';
```

- 如下几组数据在内存中如何存储?
  - 10000个学生的学习成绩
  - 200行200列矩阵
  - 8000个字符
- 这些数据的特点:
  - 具有相同的数据类型
  - 使用过程中需要保留原始数据
- ?

数组

- 属于构造数据类型
- 是同类型同性质的一组元素顺序存放构成的数据集合
- 所有数据共用同一个名字，通过下标区分不同的数据
- 处理时可通过循环控制变量控制下标的变化来批量处理数组中的数据
- 示例：

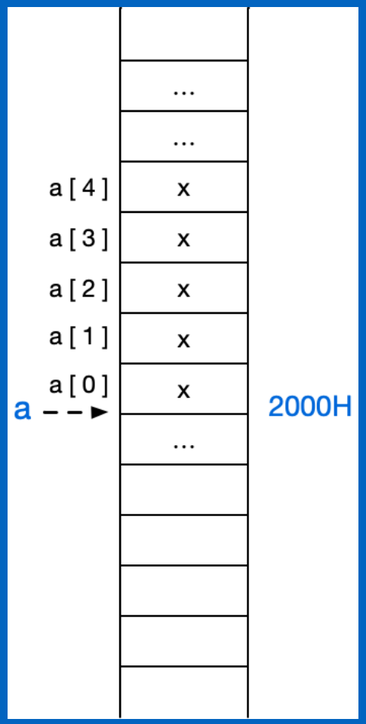
```
int cj[5];
cj[0]=75; cj[1]=85; cj[2]=72; cj[3]=88;  cj[4]=90;
```

- 对照：

```
int cj1, cj2, cj3, cj4, cj5;
cj1=75; cj2=85; cj3=72; cj4=88;  cj5=90;
```

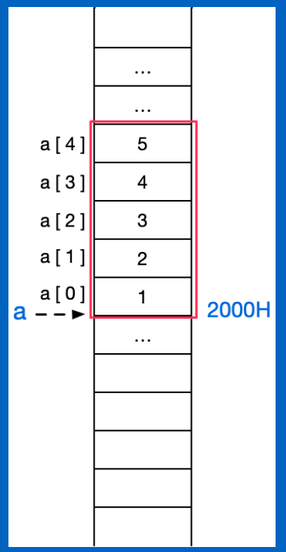
一维数组

- 数组的定义: 数据类型 数组名[常量表达式];
  - 数组名：合法的标识符
  - 常量表达式：数组的大小，即元素的个数
- 例： int a [5];
- 数组元素的使用: 数组名[下标]
  - 数组 a 有5个元素
  - a[0], a[1], a[2], a[3], a[4]
  - 下标从 0 开始
    - 第一个元素 a[0]
    - 第五个元素 a[4]



数组元素的访问

- 数组整体定义
- 元素分别使用： 不能整体引用
- 常用 for 循环
  - 循环变量初值：初始下标
  - 结束条件：结束下标
  - 循环变量变化：下标变化

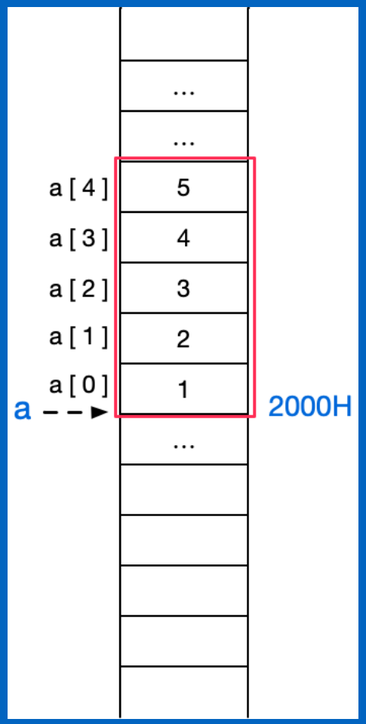



```c
#include <stdio.h>
int main (){
    int  i, a[5];

    a[0] = 1;
    a[1] = 2;
    a[2] = 3;
    a[3] = 4;
    a[4] = 5;
    //for(i = 0; i < 5; i++)
    //  a[i] = i + 1;

    //printf("%d",a); 出错
    for(i = 0; i < 5; i++)
        printf("%d", a[i]);

    return 0;
}
```

数组元素的访问

- 数组整体定义，元素分别使用

```
#include <stdio.h>
int main (){
    int  i, a[5];

    //scanf("%d",a); 出错
    for(i = 0; i < 5; i++){
        printf("Input a[%d]:", i);
        scanf("%d",&a[i]); // a[i] 可视为一个普通变量
    }

    return 0;
}
```



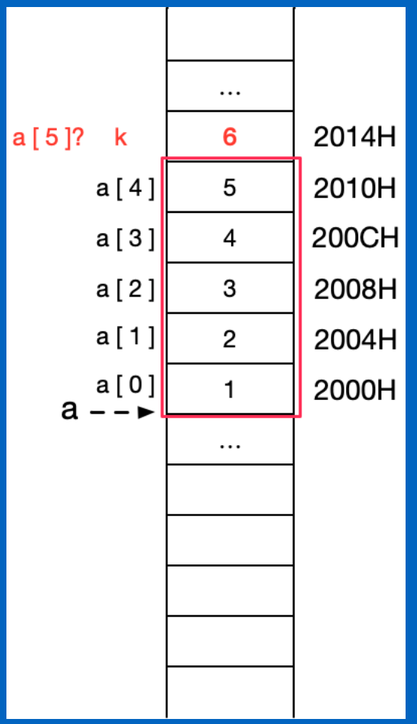
数组的大小

- 在C中，数组的大小是定长的
  - 不能定义变长的数组，为什么?
  - 常用符号常量代表数组的大小
- 需要"动态"扩展的情况：链表等
- 不允许为空
- 不允许非整型

```c
#include <stdio.h>
#define N 3
int main (){
    //符号常量指定数组的大小
    int  i, a[N], b[N];

    //定义数组的大小不能用变量
    //int m = 5;
    //int c[m];   无法通过编译
    //int n;
    //scanf("%d", &n);
    //int d[n];   无法通过编译

    a[0] = 1;
    a[1] = 2;
    a[2] = 3;

    for(i = 0; i < N; i ++)  //符号常量
        printf("%d", a[i]);

    return 0;
}
```

易错点：a [5] 的含义?

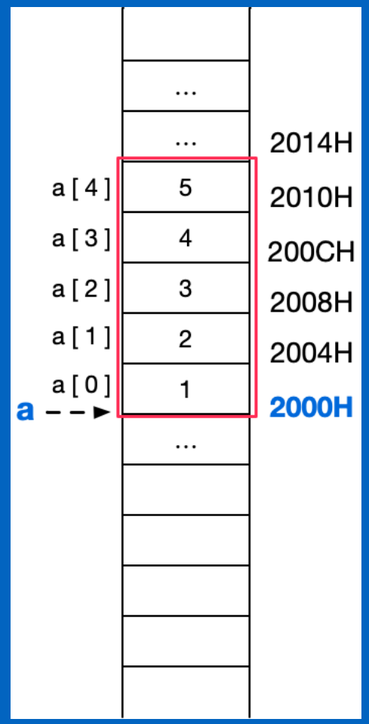
```c
#include <stdio.h>
int main (){
    int  i;
    //a[5]前面有int,表示定义5个元素的数组 a;
    int  a[5], k=10000;

    a[0] = 1;
    a[1] = 2;
    a[2] = 3;
    a[3] = 4;
    a[4] = 5;
    //以下a[5]是数组中的第6个元素
    //数组中a [5]元素本不存在
    //元素越界，应予避免
    a[5] = 6;

    for(i = 0; i < 5; i++)
        printf("%d", a[i]);

    return 0;
}
```



数组名的意义

- 数组名表示数组所占内存的首地址
- 数组名是地址常量
- 本例中：数组名 a 即是 2000H 的别名

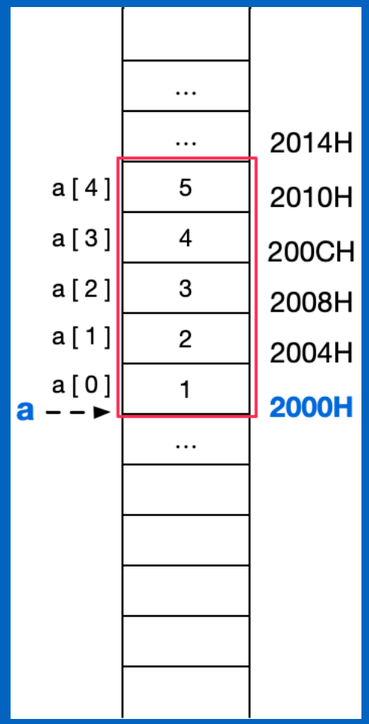
```
#include <stdio.h>
int main (){
    int  i;
    //a[5]前面有int,表示定义5个元素的数组 a;
    int  a[5];

    a[0] = 1;
    a[1] = 2;
    a[2] = 3;
    a[3] = 4;
    a[4] = 5;

    for(i = 0; i < 5; i++)
        printf("%d", a[i]);

    return 0;
}
```



数组的存储

- 数组所占内存字节数 =
  sizeof(元素数据类型) * 数组元素个数

```
//数组 a 中每个元素是int，占 4 字节
//数组 a 在内存中占 4 * 5 = 20 字节
int  a[5];
a[0] = 1; a[1] = 2; a[2] = 3; a[3] = 4; a[4] = 5;

//数组 c 中每个元素是char，占 1 字节
//数组 c 在内存中占 1 * 3 = 3 字节
char  c[3];
c[0] = 'D'; c[1] = 'H'; c[2] = 'U';

//数组 f 中每个元素是float，占 4 字节
//数组 c 在内存中占 4 * 20 = 80 字节
float  f[20];
```



数组的初始化

- 定义数组时，允许为数组的部分或全部元素赋初值
- 初值应被放在花括{ }号中
- 全部元素赋初值时，数组长度可省略
  - int a[5]={0,2,4,6,8};
  - 也可写为： int a[ ]={0,2,4,6,8};
  - 作用皆相当于：
    a[0]=0; a[1]=2; a[2]=4; a[3]=6; a[4]=8;
- 部分元素赋初值时，未被赋值元素默认为0
  - int a[10]={1,3,5,7,9};
    其中，a[5]~a[9]之间的元素皆为0
  - 数组中全部元素值为0:
    int a[10]={0,0,0,0,0,0,0,0,0,0};
    int a [10]={0};
- 避免以下错误

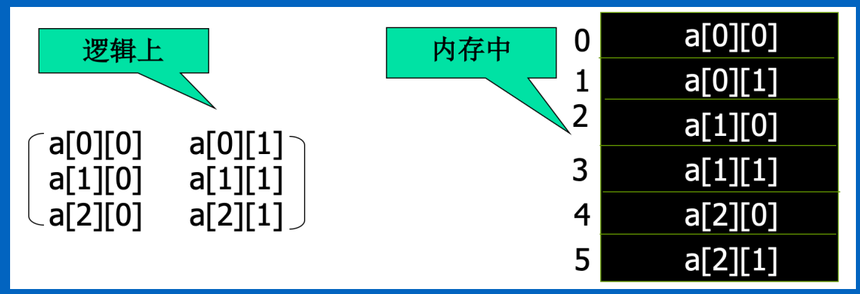
```
int  a[10];
a={1,3,5,7,9};      //数组名不能被赋值

int  a[10];
a[10]={1,3,5,7,9};  //数组不能被赋值

int c[3]={1,2, 3,4};      //初值个数超出数组长度
```

二维数组

- 定义：数据类型　数组名[行数][列数]

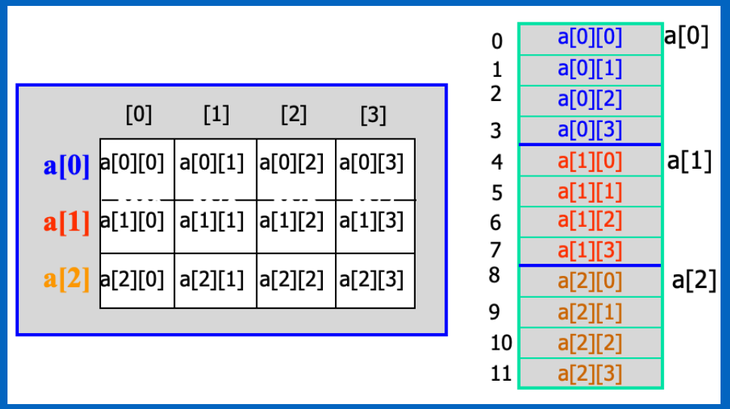
```
int a[3][2]; //3行2列
float b[2][5];
int c[2][3][4];
```

- 逻辑上是二维的
- 内存中是一维的
  - 按行存储
  - 先行后列



二维数组

- 数组的数组
- 例 int a[3][4];
  - 可视为由3个元素组成a[0],a[1],a[2]
  - 每个元素a[i] 又可视为由 包含4个元素的一维数组 组成



二维数组元素的访问

- 数组整体定义，元素分别使用
- 常用 双重 循环
  - 外循环控制行的变化
  - 内循环控制列的变化
- 例：使用二维数组存储如下数据,并按三行两列形式输出

  1 2
  3 4
  5 6

```c
#include <stdio.h>
int main (){
    int a[3][2],i,j;

    //从键盘上输入数组元素
    for(i = 0; i < 3; i++)
        for(j = 0; j < 2; j++)
            scanf("%d",&a[i][j]);

    //输出数组元素
    for(i = 0;i < 3; i++){
        for(j = 0;j < 2; j++)
            printf("%4d",a[i][j]);
        printf("\n");
    }

    return 0;
}
```

二维数组初始化

- 按行给所有元素赋初值
  - 每行数据组织在一对花括号内

```
int  a[2][3]={{1,2,3},{4,5,6}};
// 逻辑存储:
1 2 3
4 5 6
```

- 按存放顺序对所有元素赋初值

```
int  a[2][3]={1,2,3,4,5,6};
// 逻辑存储:
1 2 3
4 5 6
```

- 按行给部分元素赋初值，未被赋值的元素自动为0

```
int  b[3][4]={{1,2},{0,3,4},{0,0,5}}
// 逻辑存储:
1 2 0 0
0 3 4 0
0 0 5 0
```

- 按行赋初值也可省略第一维的长度

```
int  c[ ][3]={{1},{ },{2}};
// 逻辑存储:
1 0 0
0 0 0
2 0 0
```

数组相关常见算法

- 一维数组相关算法
  - 查找(顺序、二分)
  - 计算最大值/位置/和/平均值
  - 倒置
  - 排序（选择、冒泡）
  - 插入
  - 删除

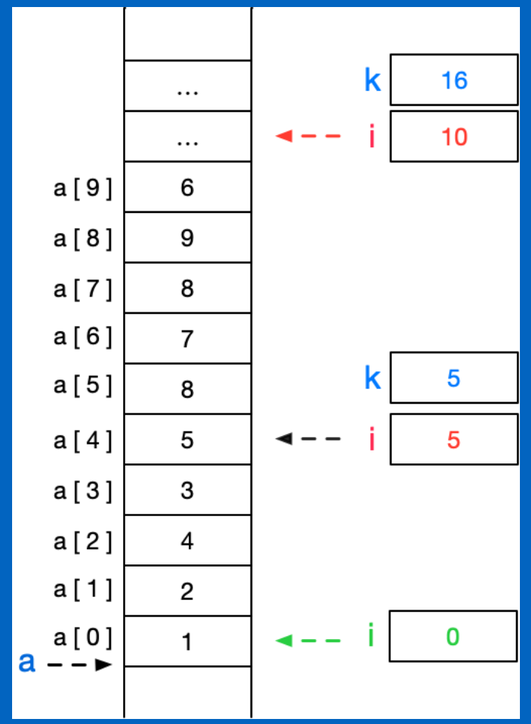
- 二维数组相关算法
  - 转置
  - 计算最大值
  - 各行各列的最大值/求和
  - 杨辉三角形
  - 上三角元素之和

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & u_{n-1,n} \\ 0 & & & & u_{n,n} \end{bmatrix}$$

顺序查找

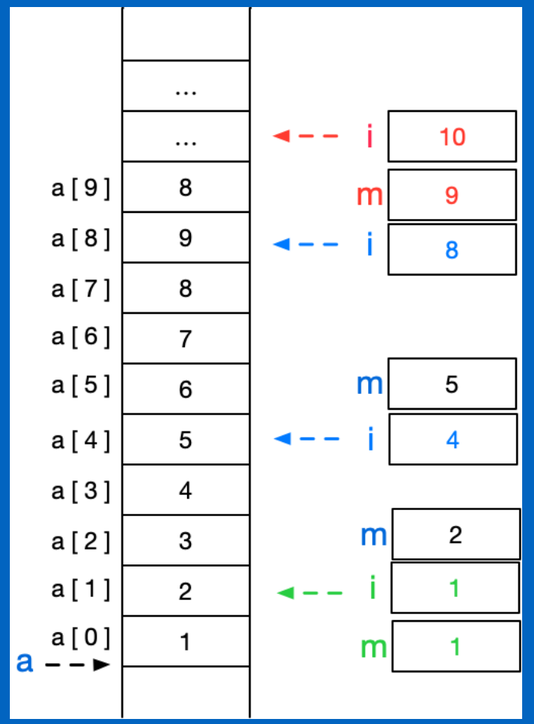
- 在数组中查找指定数据
- 如:整数数组a[10]中，查找某个数据k是否存在?

```c
#include <stdio.h>
#define MAX 100
int main (){
    int i, n, a[MAX], k;
    printf("Input data number n:");
    scanf("%d",&n);
    //输入数组
    for(i = 0; i < n; i++){
        printf("Input data %d: ", i);
        scanf("%d",&a[i]);
    }
    //输入查找数据
    printf("please input search number  ");
    scanf("%d",&k);
    //查找
    for(i = 0; i < n; i++)
        if (a[i] == k) break;
    //判断是否找到，并输出结果
    if(i >= n) printf("not found\n");
    else printf("Found. %d,%d", i, a[i]);

    return 0;
}
```



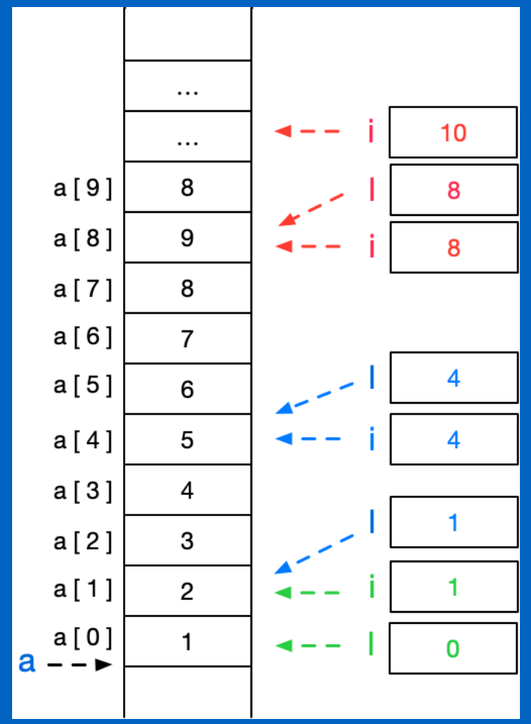
一维数组中的最大值

```c
#include <stdio.h>
#define MAX 10
int main (){
    int  a[MAX], i, m;
    printf("Input a[%d] :", MAX);

    for( i = 0; i < MAX; i++)
        scanf("%d", &a[i]);

    m=a[0];

    for( i = 1; i <= MAX-1; i++)
        if( a[i] > m)  m = a[i];

    printf("%d\n", m);

    return 0;
}
```



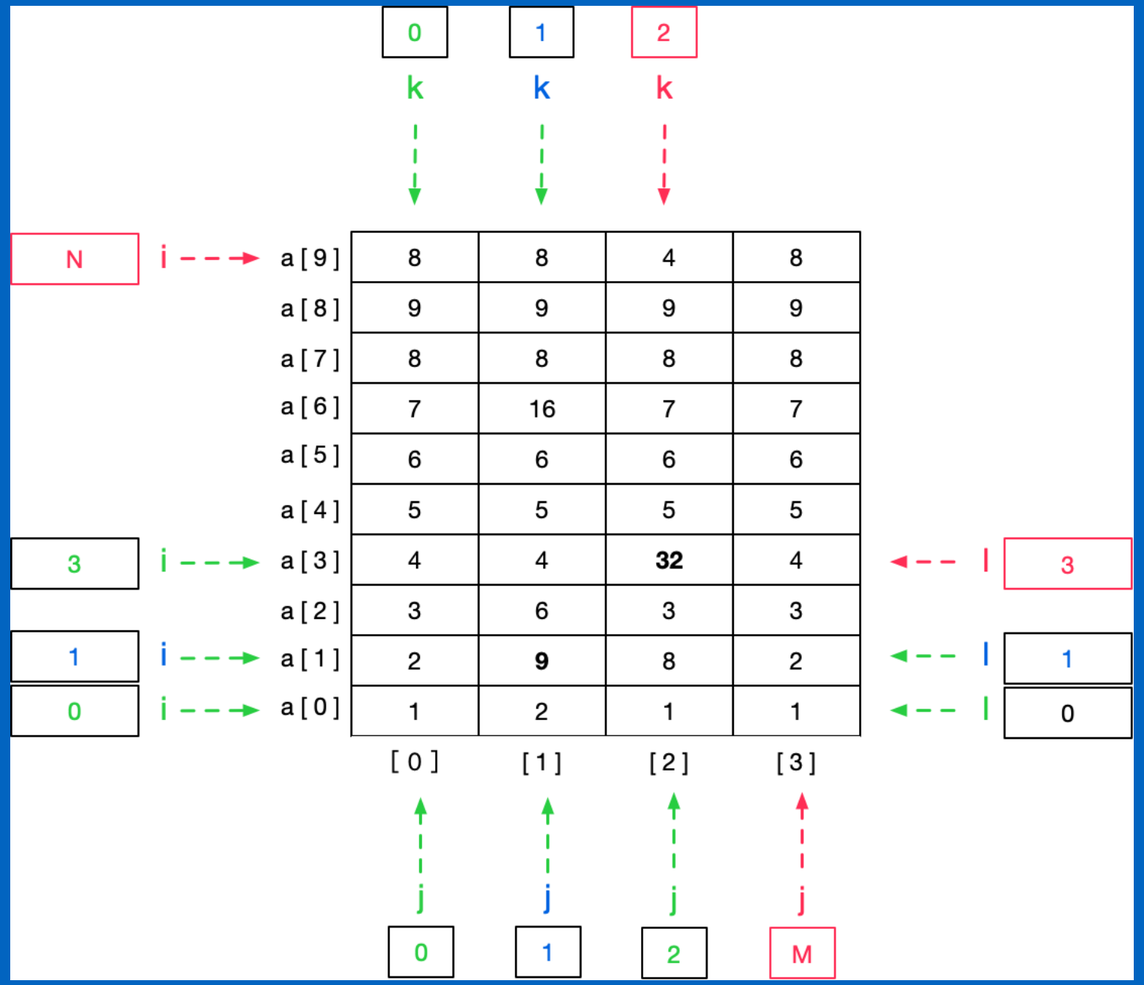
一维数组中的最大值及其所在的位置

```c
#include <stdio.h>
#define MAX 10
int main () {
    int  a[MAX], i, l;
    printf("Input a[%d] :", MAX);

    for( i = 0; i < MAX; i++)
        scanf("%d", &a[i]);

    l=0;
    for( i = 1; i <= MAX-1;  i++)
        if( a[i] > a[l] )
            l = i;

    printf("%d, %d\n", l, a[l]);

    return 0;
}
```



二维数组中的最大值及其所在的位置

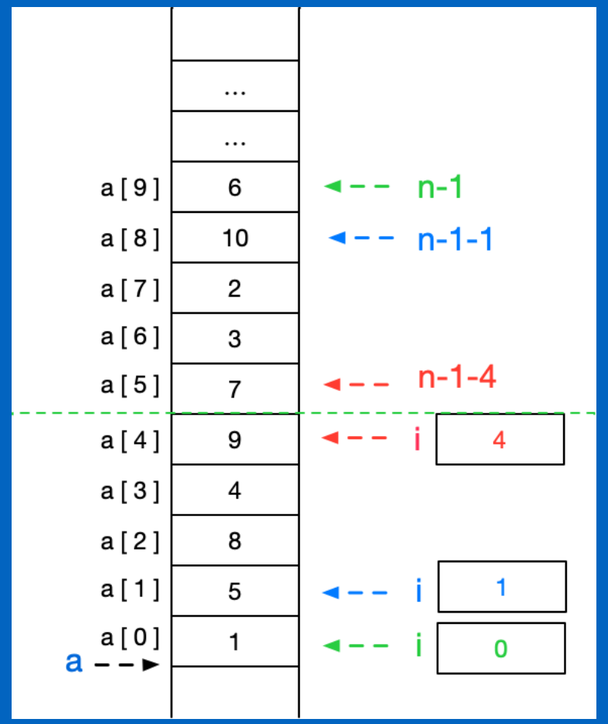
```c
#include <stdio.h>
#define MAX 10
int main (){
    int  a[MAX][MAX], n, i, j, l, k;
    printf("Input a[%d][%d]:\n", MAX, MAX);

    for( i = 0; i < MAX; i++)
        for( j = 0; j < MAX; j++)
            scanf("%d", &a[i][j]);

    l = 0;
    k = 0;
    for( i = 0; i <= MAX-1; i++)
        for( j = 0; j <= MAX-1; j++)
            if( a[i][j] > a[l][k] )  {
                l = i;
                k = j;
            }

    printf("%3d,%3d,%5d\n", l, k, a[l][k]);

    return 0;
}
```



元素倒置（一）

- 将一个数组的元素值按逆序重新存放

```c
#include <stdio.h>
#define MAX 100
int main (){
    int a[MAX],n,i,t;
    printf("please input n:");
    scanf("%d",&n);
    //输入数组a
    printf("please input 数组a ");
    for(i = 0; i < n; i++)
        scanf("%d",&a[i]);
    printf("\n");

    //交换数组
    for(i = 0; i < n/2; i++){
        t = a[i];
        a[i] = a[n-1-i];
        a[n-1-i] = t;
    }

    //输出交换后的数组
    for(i = 0; i < n; i++)
        printf("%4d",a[i]);

    return 0;
}
```



元素倒置（二）

- 将一个数组的元素值按逆序重新存放

```
#include <stdio.h>
#define MAX 100
int main (){
    int a[MAX],n,i,t;
    printf("please input n:");
    scanf("%d",&n);
    //输入数组a
    printf("please input 数组a ");
    for(i = 0; i < n; i++)
        scanf("%d",&a[i]);
    printf("\n");

    //交换数组
    for(i = 0, j = n-1;i < j; i++, j--){
        t = a[i];
        a[i] = a[j];
        a[j] = t;
    }

    //输出交换后的数组
    for(i = 0; i < n; i++)
        printf("%4d",a[i]);

    return 0;
}
```