中南大學
CENTRAL SOUTH UNIVERSITY

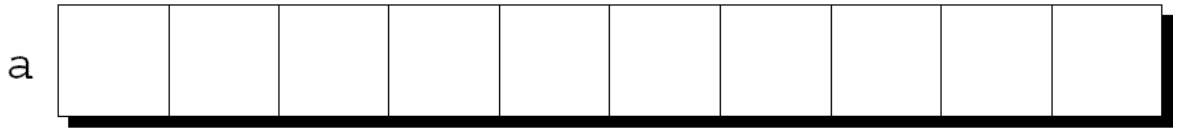
# 第八讲

# 数　　组

# 8、数组

## 8.1 一维数组

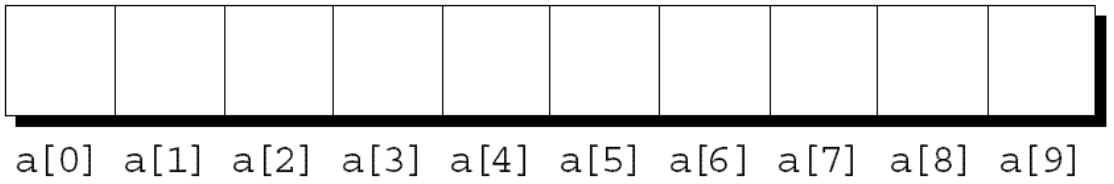
- 数组：包含**多个**数据值，每个数据值具有**相同的数据类型**
- 数据值：元素
- 一维数组：元素一个**接**一个的编排在单独一行（列）




- 数组声明：类型 数组名 数量

```
int a[10];
```

- 存取特定数组元素：数组名[下标 / 索引]
- 数组元素始终**从0开始**，长度为n的数组的索引：0~n-1



# 8、数组

## 8.1 一维数组

- 每个数组元素都可以视为一个**独立**的对应类型的**变量**使用

```
a[0] = 1;
scanf("%d\n", &a[5]);
printf("%d\n", a[5]);
++a[i];
```

- 数据元素经常和**for循环**结合使用

```
for (i = 0; i < N; i++)
  a[i] = 0;                    /* clears a */

for (i = 0; i < N; i++)
  scanf("%d", &a[i]);          /* reads data into a */

for (i = 0; i < N; i++)
  sum += a[i];                 /* sums the elements of a */
```

# 8、数组

## 8.1 一维数组

- C语言不会检查数组下标的范围

```
int a[10], i;

for (i = 1; i <= 10; i++)
    a[i] = 0;
```

  - 以上可能会产生无限循环

- 数组下标也可以是任何的整数表达式

```
a[i+j*10] = 0;
a[i++]
```

- 示例：录入一串数，反向输出

```
/*reverse.c Reverses a series of numbers */
#include <stdio.h>
#define N 10
int main(void) {
  int a[N], i;
  printf("Enter %d numbers: ", N)
  for (i = 0; i < N; i++)
    scanf("%d", &a[i]);
  printf("In reverse order:");
  for (i = N - 1; i >= 0; i--)
    printf(" %d", a[i]);
  printf("\n");
  return 0;
}
```

```
Enter 10 numbers: 34 82 49 102 7 94 23 11 50 31
In reverse order: 31 50 11 23 94 7 102 49 82 34
```

# 8、数组

## 8.1 一维数组

- 数组的**初始化**

```
int a[10] = {1, 2, 3, 4, 5, 6};
/* initial value of a is {1, 2, 3, 4, 5, 6, 0, 0, 0, 0} */
```

- 如果初始化式比数组短，那么数组中剩余的元素**赋值为0**

```
int a[10] = {0};不能为空
/* initial value of a is {0, 0, 0, 0, 0, 0, 0, 0, 0, 0} */
```

- 初始化式不能比数组长
- 如果给定了初始化式，可以**省略**数组长度

```
int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

- 指定初始化式

```
int a[15] ={0, 0, 29, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 48};
int a[15] = {[2] = 29, [9] = 7, [14] = 48};
```

## 8.1 一维数组

- 示例
  - 检查数中是否出现多于一次的数字
  - 输出Repeated digit or No repeated

```c
/* Checks numbers for repeated digits */
#include <stdbool.h> /* C99 only */
#include <stdio.h>
int main(void)  {
    bool digit_seen[10] = {};
    int digit;
    long n;
    printf("Enter a number: ");
    scanf("%ld", &n);
    while (n>0){
        digit = n % 10;
        if (digit_seen[digit]) break;
        digit_seen[digit] = true;
        n /= 10;
    }
    if (n > 0)
        printf("Repeated digit\n");
    else
        printf("No repeated digit\n");
    return 0;
}
```

# 8、 数 组

## 8.1 一维数组

- 对数组使用sizeof运算符
  - 返回数组的大小（总字节数）
  - 如果数组a有10个整数，那么sizeof(a)通常等于40
  - 数组的长度=数组的大小/数组元素大小
  - sizeof(a)/sizeof(a[0])

```
for (i = 0; i < sizeof(a) / sizeof(a[0]); i++)
  a[i] = 0;
```

```
for (i = 0; i < (int) (sizeof(a) / sizeof(a[0])); i++)
  a[i] = 0;
```

```
#define SIZE ((int) (sizeof(a) / sizeof(a[0])))

for (i = 0; i < SIZE; i++)
  a[i] = 0;
```

## 8.1 一维数组

### 示例：计算利息

- 显示在几年内100元投资在不同利率下的价值
- 输入：利率，要投资年数
- 输出：投资总价值每年计算一次，表格将显示出在输入利率和紧随其后的4个更高的利率下投资的总价值

```
Enter interest rate: 6
Enter number of years: 5

Years      6%       7%       8%       9%      10%
  1     106.00   107.00   108.00   109.00   110.00
  2     112.36   114.49   116.64   118.81   121.00
  3     119.10   122.50   125.97   129.50   133.10
  4     126.25   131.08   136.05   141.16   146.41
  5     133.82   140.26   146.93   153.86   161.05
```

# 8、数组

```c
/* Prints a table of compound interest */
#include <stdio.h>
#define NUM_RATES((int) (sizeof(value) / sizeof(value[0])))
#define INITIAL_BALANCE 100.00
int main(void){
    int i, low_rate, num_years, year;
    double value[5];
    printf("Enter interest rate: ");
    scanf("%d", &low_rate);
    printf("Enter number of years: ");
    scanf("%d", &num_years);
    printf("\nYears");
    for (i = 0; i < NUM_RATES; i++){
        printf("%6d%%", low_rate + i);
        value[i] = INITIAL_BALANCE;
    }
    printf("\n");
    for (year = 1; year <= num_years; year++){
        printf("%3d ", year);
        for (i = 0; i < NUM_RATES; i++) {
            value[i] += (low_rate + i) / 100.0 * value[i];
            printf("%7.2f", value[i]);
        }
        printf("\n");
    }
    return 0;
}
```

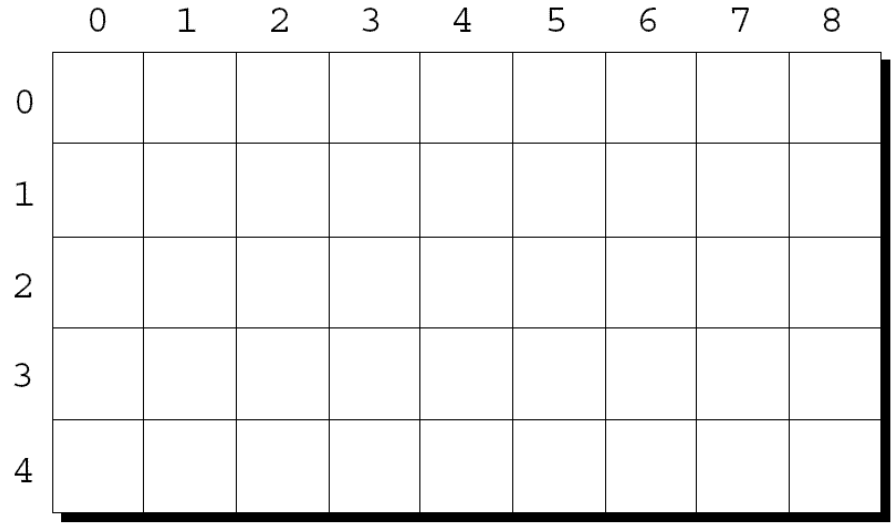
## 8.2 多维数组

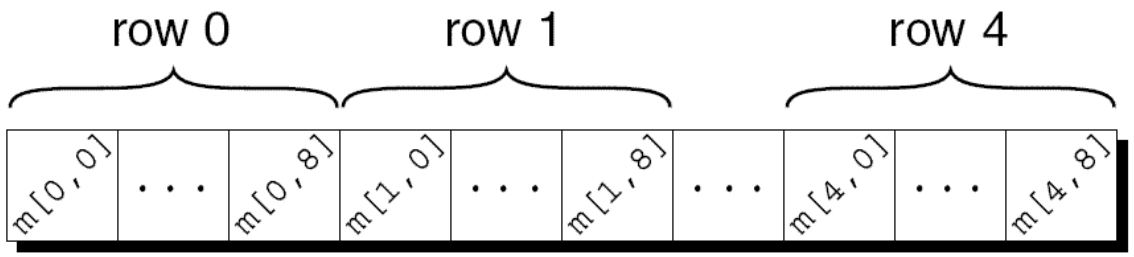
- 数组可以有**任意维数**
- **二维**数组（矩阵）

```
int m[5][9];
```

访问i行j列：m[i][j]

- 不能写成 m[i,j] ，
  会被认为是逗号操作符，所以m[i,j]
  等同于 m[j]
- 在内存中，C语言按照行主序（行优先）存储数组



## 8.2 多维数组

- 嵌套for循环访问数组

```
#define N 10
double ident[N][N];
int row, col;
for (row = 0; row < N; row++)
  for (col = 0; col < N; col++)
    if (row == col)
      ident[row][col] = 1.0;
    else
      ident[row][col] = 0.0;
```

## 8.2 多维数组

- 初始化
  - 嵌套一维初始化式
  - 不够的，0 填充
  - 后两行赋值为0
  - 剩余的都为0
  - 省略内层花括号
  - C99：指定初始化式

```
double ident[2][2] = {[0][0] = 1.0, [1][1] = 1.0};
```

```
int m[5][9] = {{1, 1, 1, 1, 1, 0, 1, 1, 1},
               {0, 1, 0, 1, 0, 1, 0, 1, 0},
               {0, 1, 0, 1, 1, 0, 0, 1, 0},
               {1, 1, 0, 1, 0, 0, 0, 1, 0},
               {1, 1, 0, 1, 0, 0, 1, 1, 1}};
```

```
int m[5][9] = {{1, 1, 1, 1, 1, 0, 1, 1, 1},
               {0, 1, 0, 1, 0, 1, 0, 1, 0},
               {0, 1, 0, 1, 1, 0, 0, 1, 0}};
```

```
int m[5][9] = {{1, 1, 1, 1, 1, 0, 1, 1, 1},
               {0, 1, 0, 1, 0, 1, 0, 1},
               {0, 1, 0, 1, 1, 0, 0, 1},
               {1, 1, 0, 1, 0, 0, 0, 1},
               {1, 1, 0, 1, 0, 0, 1, 1, 1}};
```

```
int m[5][9] = {1, 1, 1, 1, 1, 0, 1, 1, 1,
               0, 1, 0, 1, 0, 1, 0, 1, 0,
               0, 1, 0, 1, 1, 0, 0, 1, 0,
               1, 1, 0, 1, 0, 0, 0, 1, 0,
               1, 1, 0, 1, 0, 0, 1, 1, 1};
```

# 8、数组

## 8.2 多维数组

### 常量数组

```
const char hex_chars[] =
        {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
         'A', 'B', 'C', 'D', 'E', 'F'};
```

### 示例

- 发牌程序
- 输入：牌的数量n，输出：n张随机不重复的牌

```
/* Deals a random hand of cards */
#include <stdbool.h> /* C99 only */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define NUM_SUITS 4
#define NUM_RANKS 13
int main(void) {
   bool in_hand[NUM_SUITS][NUM_RANKS] = {};
   int num_cards, rank, suit;
   const char rank_code[] = {'2','3','4','5','6','7','8',
                             '9','t','j','q','k','a'};
   const char suit_code[] = {'c','d','h','s'};
   srand((unsigned) time(NULL));
   printf("Enter number of cards in hand: ");
   scanf("%d",&num_cards);
   printf("Your hand:");
   while (num_cards >0) {
      suit = rand() % NUM_SUITS; /* picks a random suit */
      rank = rand() % NUM_RANKS; /* picks a random rank */
      if (!in_hand[suit][rank]) {
          in_hand[suit][rank] = true ;
          num_cards --;
          printf(" %c%c", rank_code[rank], suit_code[suit]);
      }
    }
    printf("\n");
    return 0;
}
```

## 8.3 变长数组

- 变长数组的长度是在程序执行时计算的，而不是在编译时计算的
- 变长数组的长度可以是表达式

```
int a[3*i+5];
int b[j+k];
```

- 也可以是多维的

```
int c[m][n];
```

- 变长数组没有初始化

```
/* Reverses a series of numbers using a varia
 array - C99 only */
#include <stdio.h>
int main(void) {
  int i, n;
  printf("How many numbers do you want to rev
  scanf("%d", &n);
  int a[n]; /* C99 only - length of array dep
  printf("Enter %d numbers: ", n);
  for (i = 0; i < n; i++)
     scanf("%d", &a[i]);
     printf("In reverse order:");
     for (i = n - 1; i >= 0; i--)
     printf(" %d", a[i]);
     printf("\n");
     return 0;
}
```