

# 目 录

<b>第一章 流程控制语句</b>	<b>1</b>
1.1 逻辑值 . . . . .	1
1.2 关系运算符和关系表达式 . . . . .	1
1.3 逻辑运算符和逻辑表达式 . . . . .	1
1.4 多分支选择语句 . . . . .	2
1.5 循环语句 . . . . .	4



# 1

## 流程控制语句

### 1.1 逻辑值

- ▶ 关系表达式和逻辑表达式都会得到一个逻辑值
- ▶ 0 代表假，1 表示真 (任意非零值)
- ▶ 关系表达式和逻辑表达式只会为 0 或者 1

### 1.2 关系运算符和关系表达式

- ▶ 6 种关系运算符：<、<=、>、>=、==、!=
- ▶ 关系运算符只有两种可能，0 或者 1

### 1.3 逻辑运算符和逻辑表达式

- ▶ 逻辑与"&&"
- ▶ 逻辑或"||"
- ▶ 逻辑非"! "
- ▶ 优先级：非 > 与 > 或
- ▶ 逻辑运算符两边可以是任意合法表达式

a	b	!a	!b	a&&b	a  b
非0	非0	0	0	1	1
非0	0	0	1	0	1
0	非0	1	0	0	1
0	0	1	1	0	0

## 1.4 多分支选择语句

算法 1-1 加减乘除的运算

```

1  #include <stdio.h>
2  int main() {
3      int num1, num2;
4      char op;
5      printf("请输入两个整数和一个运算符 (如 3 + 5) : ");
6      scanf("%d%c%d", &num1, &op, &num2);
7      switch (op) {
8          case '+':
9              printf("%d + %d = %d\n", num1, num2, num1 + num2);
10             break;
11          case '-':
12              printf("%d - %d = %d\n", num1, num2, num1 - num2);
13             break;
14          case '*':
15              printf("%d * %d = %d\n", num1, num2, num1 * num2);
16             break;
17          case '/':
18              if (num2 != 0) {
19                  printf("%d / %d = %d\n", num1, num2, num1 / num2);
20              } else {
21                  printf("错误: 除数不能为 0! \n");
22              }
23             break;
24          default:
25              printf("无效的运算符! \n");
26             break;
27      }
28      return 0;
29  }

```

算法 1-2 成绩分段评定

```
1 #include <stdio.h>
2 int main() {
3     int score;
4     printf("请输入成绩 (0-100) : ");
5     scanf("%d", &score);
6     if (score < 0 || score > 100) {
7         printf("无效的成绩! \n");
8         return 1; // 结束程序
9     }
10    switch (score / 10) {
11        case 10: // 100 分
12        case 9: // 90-99 分
13            printf("成绩等级: A\n");
14            break;
15        case 8: // 80-89 分
16            printf("成绩等级: B\n");
17            break;
18        case 7: // 70-79 分
19            printf("成绩等级: C\n");
20            break;
21        case 6: // 60-69 分
22            printf("成绩等级: D\n");
23            break;
24        default: // 0-59 分
25            printf("成绩等级: F\n");
26            break;
27    }
28    return 0;
29 }
```

- ▶ 如果运算符不匹配任何 case，则执行 default 块，提示用户输入的运算符无效。
- ▶ switch(), 括号内输入需要判断的分支变量。
- ▶ case”: 中用于选择的常量只能是数值型，单个字符也是数值型，用单引号括住，并且加一个分号。
- ▶ case 后面不能跟一个范围，或者运算，只能跟一个数值。
- ▶ break 用于跳出分支选择语句。

## 1.5 循环语句

for 循环

算法 1-3 计算 1-100 中偶数的和

```
1 #include <stdio.h>
2 int main() {
3     int sum = 0;
4     for (int i = 1; i <= 100; i++) {
5         if (i % 2 == 0) {
6             sum += i;
7         }
8     }
9     printf("1到100之间所有偶数的和为: %d\n", sum);
10    return 0;
11 }
```

while 循环

算法 1-4 计算 1-100 中偶数的和

```
1 #include <stdio.h>
2 int main() {
3     int sum = 0;
4     int i = 1; // 初始化循环变量
5     while (i <= 100) { // 循环条件
6         if (i % 2 == 0) { // 判断是否为偶数
7             sum += i; // 累加偶数
8         }
9         i++; // 更新循环变量
10    }
11    printf("1到100之间所有偶数的和为: %d\n", sum);
12    return 0;
13 }
```

do while 循环

算法 1-5 计算 1-100 中偶数的和

```
1 #include <stdio.h>
2 int main() {
3     int sum = 0;
4     int i = 1; // 初始化循环变量
5     do {
```

```
6         if (i % 2 == 0) { // 判断是否为偶数
7             sum += i; // 累加偶数
8         }
9         i++; // 更新循环变量
10    } while (i <= 100); // 循环条件
11    printf("1到100之间所有偶数的和为: %d\n", sum);
12    return 0;
13 }
```