

1.ASCII码补充

1) 把字符转换成数值

2) 特点:

1》区分大小写

2》没有中文

3》字符在运算是会自动转换成ASCII码，所有的字符在计算机的世界里都是二进制，如果不统一，同一个字符在不同的电脑上会显示不同的二进制，所以，就需要ASCII码来统一，我国有GBK2312和UTF-8两种标准

2.if的使用（选择结构）

1) 用法

1》if()

```
{
```

```
}
```

2》if()

```
{
```

```
}else
```

```
{
```

```
}
```

3》if()

```
{
```

```
}else if()
```

```
{
```

```
}else if()
```

```
{
```

```
}.....
```

注意：只要有一个满足，后面的不执行

4》if()

```
{
```

```
}else if()
```

```
{
```

```
}else
```

```
{
```

```
}
```

注意：只要有一个满足，后面的不执行,但若是都不满足，就执行else里的语句

5》if();

注意：只对if后面的第一个语句有效，以分号（；）结尾。

空格没有影响，只要有分号，就结束

2)特点

1》一次if语句只能执行给出的所有操作中的一个

2》可嵌套：效率更高

```
int a=3;
if (a>1) {
    printf("a>1\n");
    if (a>2) {
        printf("a>2\n");
        if(a++){
            printf("a++\n");
        }
    }
}
```

3》if () ; 不能定义变量

4》除了if();这种用法，其它用法if()后面不能有分号

5》if(){}中，如果条件判断中，有常量等于判断，建议常量写在前面

注意：if(a == 2)

{

}

这是对的，不要少写个等号，变成赋值

3.switch语句

1》作用：根据不同的匹配值执行不同的操作

2》格式

```
int a=0;
switch (a) {
    case 1:
        printf("a=1\n");
        break;
    case 2:
        printf("a=2\n");
        break;
    case 3:
        printf("a=3\n");
        break;
    case 4:
        printf("a=4\n");
        break;
    default:
        printf("没有匹配的int值\n");
        break;
}
```

3》特点

1.条件结果只能是整数,而case只能是整型常量.注意:case 'a';

2.break 可有可无,有则执行完跳出,没有则继续执行,直到遇到下一个break或执行完.

3.default可有可无,有则如果条件都不满足,会执行这句,无则条件都不满足,执行结束.

4.case下面要定义变量必须加{}.且case后的常量值不能重复

5.switch语句可以全部用if语句替代.但if语句不能全用switch替代.

4.do while与while

1》格式:

while(<条件>) {<语句>;} do {<语句>} while(<条件>);

2》两者区别

do while中的语句至少执行一次，而while中的语句，如果条件不满足，就一次有不执行

3》特点

- 1.只有满足条件循环才会执行.
- 2.注意死循环的出现.会一直循环执行下去,除非程序强制退出. **control+c**
- 3.如果碰到continue,则跳出本次循环继续下一次判断,如果条件成立继续执行循环.
- 4.如果碰到break无论条件满足与否,循环立刻结束.
- 5.continue与break跳出的都是当前所在的循环作用域.如果外层有循环,不受影响.
- 6.do while至少执行一次.而while不一定.

5.for

1》作用：根据条件反复执行一段代码

2》用法

```
/*for循环的组成.由三个语句+循环体
for(语句1;语句2;语句3){
```

循环体:需要反复执行的代码...

```
}
```

执行顺序:语句1-->判断语句2是否成立--成立-->循环体-->语句3-->判断语句2是否成立--成立-->循环体-->语句3-->再判断语句2...直到语句2不成立.for循环终止.

1.for循环第一个语句:只会执行一次:一般用于初始化变量.且如果是定义同一类型的变量,可以定义多个.方法:int a=1,b=2;

```
for (int a=0, b=2; a<10; a++) {
printf("当前是第%d行Hello world!\n",a);
}
```

2.for循环第二个语句:是一个条件语句,控制for循环是否要继续执行.如果这个条件不满足,for循环结束了.

注意:这个必须要有一个判断结果.如果为空,表示默认为真.则是死循环

```
for (int a=0; ; a++) {
printf("当前是第%d行Hello world!\n",a);
}
```

3.for循环第三个语句不一定会执行.一般用于控制循环增量或变化.

注意:不能放变量定义语句.

```
for (int a=0; a<10; int b=0) {
printf("当前是第%d行Hello world!\n",a);
}
```

4.for循环的循环体:没有限制,可以放任意你想执行的语句.

3》特点

满足条件才会执行程序

如果出现死循环,会一直执行,可用control + c强制退出

出现continue,跳出本次循环进行下次判断

出现break,不管条件满不满足,循环马上结束

continue和break只对自己所在的循环有影响,对外循环无影响

6.for循环练习题

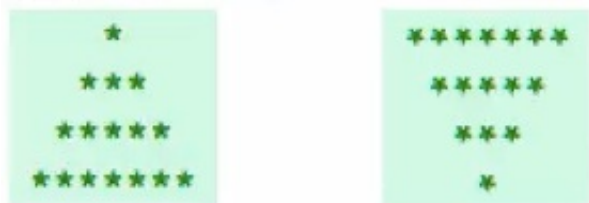
1》打印九九乘法表

```
#include <stdio.h>
int main()
{
    for (int a = 1; a <= 9; a++)
    {
        for (int b = 1; b <= a ; b++)
        {
            printf("%d * %d = %d\t",b ,a ,a * b );
        }
        printf("\n");
    }
    return 0;
}
```

运行结果为：

1 * 1 = 1									
1 * 2 = 2	2 * 2 = 4								
1 * 3 = 3	2 * 3 = 6	3 * 3 = 9							
1 * 4 = 4	2 * 4 = 8	3 * 4 = 12	4 * 4 = 16						
1 * 5 = 5	2 * 5 = 10	3 * 5 = 15	4 * 5 = 20	5 * 5 = 25					
1 * 6 = 6	2 * 6 = 12	3 * 6 = 18	4 * 6 = 24	5 * 6 = 30	6 * 6 = 36				
1 * 7 = 7	2 * 7 = 14	3 * 7 = 21	4 * 7 = 28	5 * 7 = 35	6 * 7 = 42	7 * 7 = 49			
1 * 8 = 8	2 * 8 = 16	3 * 8 = 24	4 * 8 = 32	5 * 8 = 40	6 * 8 = 48	7 * 8 = 56	8 * 8 = 64		
1 * 9 = 9	2 * 9 = 18	3 * 9 = 27	4 * 9 = 36	5 * 9 = 45	6 * 9 = 54	7 * 9 = 63	8 * 9 = 72	9 * 9 = 81	

2》打印以下图形（正三角和倒三角）



```
#include <stdio.h>
int main()
{
    for (int a=0; a<4; a++) {
        for (int b=0; b<=a+3; b++) {
            if (b<3-a) {
                printf(" ");
            }else
                printf("*");
        }
        printf("\n");
    }

    for (int a=0; a<4; a++) {
        for (int b=0; b<7-a; b++) {
            if (b<a) {
                printf(" ");
            }else
                printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

运行结果为：

```
 *
***
*****
*****
*****
*****
***
 *
```