putchar(c) // 输出一个字符

c = getchar() // 获取一个字符

格式输出printf

int a = 123;

int b = 12345;

printf("%4d, %4d", a, b);（如果数据的长度小于4，则左端补空格，如果大于4，按实际位数输出）

输出结果为：空格123，,12345

int c = -1;

printf("%d, %o\n", c, c);

printf("%d, %x\n", c, c);

输出结果为-1,37777777777

-1, ffffffff

原因：%o是按照八进制整数输出，是将内存单元中的各位的值（0或1）按照八进制的形式输出，输出不带符号。-1的补码为11 111 111 111 111 111 111 111 111 111 111，转换为八进制就是：37777777777。

同理%x是按照十六进制输出，-1的补码转换为十六进制就为ffffffff

%u：以无符号数输出

int d = -2;

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

输出为：-2,4294967294

原因：-2的补码为1111 1111 1111 1111 1111 1111 1111 1110，按照无符号数计算就是4294967294。

%s：输出字符串

printf("%s, %7.2s\n", "CHINA", "CHINA");

输出结果：CHINA, CH（中间补了5个空格）

%7.2s：输出占7列，但只取字符串左端的前2个字符，左补空格。

%f：输出实数，以小数形式输出。

printf("%6.2f\n", 12.3456);

输出为: 12.35（左端补了一个空格，输出的数据长度为5）

%6.2f：输出数据占6列，小数点后保留2位小数（四舍五入）

%e：以指数形式输出实数

printf("%e\n", 123.456);

输出的结果为：1.234560e+002

printf("%.2e\n", 123.456);

输出的结果为：1.23e+002

%g：用来输出实数，根据数值的大小自动选择f格式或e格式（选择输出占宽度较小的一种），且不输出无意义的0.

printf("%g\n", 123.23);

输出为：123.23（选择了f格式，且不输出之后的无意义的0）

输入：scanf

int a;

// char\* s = NULL; // 不能定义一个字符指针，然后从控制台输入

char s[3];

scanf("%d%s", &a, s); // 输入字符数组s时，只能输入两个字符，要考虑到结束字符标志’\0’

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

printf("%s\n", s);