

第五讲——补充

标准库函数

Standard Library Functions

printf sqrt scanf
isdigit Standard memcpy
calloc Library functions strcat
malloc mktime
assert

第5章-补充 标准库函数

学习要点

常用的标准库函数

随机数简介

*标准库函数

标准函数

- C语言的编译系统以标准函数的方式实现了大量常用的功能。
- 标准函数的函数原型在相应的标准库头文件（*.h）中说明；函数定义以目标码的形式保存在函数库中（用户使用时，用#include引用相应的.h文件，编译时以编译选项的方式指明需要链接的库函数）。
- C89中预定义了15个.h文件（参阅Page93，表5-1）

```
#include<stdio.h>
#include<math.h>
int main(){
    double a=3, b=4;
    double c;
    c = sqrt(a+b);
    printf("%.2f\n", c);
    return 0;
}
```

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    char str[] = "123456";
    int a;
    a = atoi(str);
    printf("%d\n",a);
}
```

```
#include<stdio.h>
#include<ctype.h>
#include<string.h>
#define N 20

int main(){
    char s[N]="buaa2020!";
    char t[N];
    int i;
    strcpy(t, s);
    for(i = 0; i <= strlen(s); i++)
        if(isalpha(s[i]))
            s[i] = toupper(s[i]);
    printf("转换前: %s\n",t);
    printf("转换后: %s\n",s);
    return 0;
}
```

*C89的标准库头文件

头文件	宏及函数的功能类别		头文件	宏及函数的功能类别
assert.h	运行时的断言检查		signal.h	事件信号处理
ctype.h	字符类型和映射		stdarg.h	变长参数表处理
errno.h	错误信息及处理		stddef.h	公用的宏和类型
float.h	对浮点数的限制		stdio.h	数据输入输出
limits.h	编译系统实现时的限制		stdlib.h	不属于其它类别的常用函数
locale.h	建立与修改本地环境		string.h	字符串处理
math.h	数学函数库		time.h	日期和时间
setjmp.h	非局部跳转			

*标准库函数: I/O (include<stdio.h>)

常用I/O库函数 (主要文本的 I/O)

函数原型	功能说明	备注
int getchar()	返回从标准输入文件中读入的一个字符	读到文件尾时返回EOF
int putchar(int c)	向标准输出文件中写入一个字符	函数出错时返回EOF
char *gets(char *buf)	从标准输入文件中读入的一行字符, 并将其保存在buf所指向的存储区	当读到文件尾部或函数出错时返回NULL, 否则返回存储区buf
char *fgets(char *buf, int n, FILE *fp)	从指定文件中读入的一行不超过n-1个字符的字符串, 并将其保存在buf所指向的存储区	当读到文件尾部或函数出错时返回NULL, 否则返回存储区buf
int puts(const char *string)	向标准输出文件中写入的一行字符, 并将字符末尾处的结束符'\0'替换成换行符'\n'	函数出错时返回EOF
int scanf(const char *format [,argument]...)	根据format中格式规定从标准输入文件中读入数据, 并保存到由argument指定的存储单元	当读到文件尾部或函数出错时返回EOF, 否则返回读入数据个数
int printf(const char *format [,argument]...)	根据format中的格式规定将argument指定的数据写入标准输出文件中	返回输出字符数, 函数出错时返回负值

*标准库函数: I/O (include<stdio.h>)

printf()函数进一步说明:

- 函数原型: int printf(const char *format[, argument] ...);
- 说明: format是一个字符串 (格式说明), argument 是数据参数 (可选)。format中可包含普通的字符串, 也可包含由%引导的说明字段 (与后面的参数按顺序一一对应, 说明对应参数的类型与输出格式)。
- 语法示例:
%[flags] [width] [.precision] type

标志	作用	默认效果	注释
-	数据在字段宽度内左对齐	右对齐	
+	在有符号数据类型前加符号	只对负数加符号	
0	数据在字段宽度内加前导0	不加前导0	
#	对类型o,x,X, 分别加前缀0,0x和0X	不加前缀	对类型c,d,i,u,s 无效
	对类型e,E,f,g,G,强制输出小数点	只有小数时输出小数点	

```
double x = 1.23456;  
int a = 789;  
printf("%.3f, %+08d, %+-8d!\n", x, a, a);
```

输出结果 :
1.235, +0000789, +789 !

*标准库函数:字符判断 (include<ctype.h>)

字符类型判断函数

函数原型	函数功能	函数原型	函数功能
int isalnum(int c)	c是否是字母或数字	int isprint(int c)	c是否是可打印字符(0x20~0x7e)
int isalpha(int c)	c是否是字母(a~z,A~Z)	int ispunct(int c)	c是否是符号, 可打印但非字母数字
int iscntrl(int c)	c是否是控制符(0x00~0x1f,0x7f)	int isspace(int c)	c是否是空白符(0x09~0x0D,0x20)
int isdigital(int c)	c是否是数字	int isupper(int c)	c是否是大写字母(A~Z)
int isgraph(int c)	c是否是可打印字符(空格除外)	int isxdigit(int c)	c是否是16进制数字(0~9, a~f, A~F)
int islower(int c)	c是否是小写字母(a~z)		

*标准库函数:字符判断 (include<ctype.h>)

字符类型判断函数

isprint()	isgraph()	isxdigit()	isdigit()	0~9	
			A~F,a~f		
		isalnum()	isalpha()	isupper()	A~Z
				islowwe()	a~z
			isdigit()	0~9	
		ispunct()	!@#\$%^&*()_-= \\{ }::'"<>?/~`		
	空格符(' ',0x20)				
isspace()	空格符(0x20), 换页符(\\f'),换行符(\\n'),回车符(\\r'),水平制表符(\\t'),垂直制表符(\\v')				
iscntrl()	控制字符(0x00~0x1f,0x7f)				

*标准库函数:字符判断 (include<ctype.h>)

字符类型判断函数典型应用

```
// 判断一个字符是否为数字型字符
int isdigit(int c){
    return c >= '0' && c <= '9';
}
```

```
// 判断一个字符是否为小写字母
int islower(int c) {
    return c >= 'a' && c <= 'z';
}
```

```
int toupper(int c); //小写字母to大写
int tolower(int c); //大写字母to小写
.....
```

应用
实例

```
#include<stdio.h>
#include<ctype.h>

int main(){
    char a = '6';
    printf("%d\n%d\n%c\n",
        isdigit(a), a, a);

    a = 'x';
    printf("%d\n%d\n%c\n",
        isdigit(a), a, a);

    a = 56;
    printf("%d\n%d\n%c\n",
        isdigit(a), a, a);

    return 0;
}
```

输出结果:

1
54
6
0
120
x
1
56
8

*标准库函数: 字符串处理 (include<string.h>)

字符串处理函数

函数原型	函数功能	函数原型	函数功能
char *strcat(char *dst, char *src)	将src追加到dst之后	int strcmp(char *s1, char *s2)	比较字符串s1和s2
char *strncat(char *dst, char *src, size_t n)	将src中前n个字符追加到dst之后	int strncmp(char *s1, char *s2, size_t n)	比较字符串s1和s2的前n个字符
char *strcpy(char *dst, char *src)	将src复制到dst中	char *strchr(char *str, int c)	在str中查找c首次出现的位置
char *strncpy(char *dst, char *src, size_t n)	将src中前n个字符复制到dst中	char *strrchr(char *str, int c)	在str中查找c最后一次出现的位置
size_t strlen(char *str)	返回字符串str长度	char *strstr(char *str, char *s1)	在str中查找s1首次出现的位置

(说明: 字符串处理跟数组密切相关, 该部分内容需要先了解数组的相关知识)

*标准库函数:通用数据处理 (include<stdlib.h>)

通用数据处理函数

函数原型	函数功能	函数原型	函数功能
int abs(int n);	n的绝对值	int rand();	生成伪随机数
double atof(char *s);	将s转换为double类型的数	void srand(unsigned int s);	设置随机数的种子
int atoi(char *s);	将s转换为int类型的数	void exit(int status);	终止程序运行
void *bsearch(void *key,void *base,size_t num,size_t width,int (*f)(void *e1, void *e2));	以二分查找的方式在排序数组base中查找元素key	void qsort(void *base,size_t num,size_t width,int (*f)(void *e1, void *e2));	以快速排序方式对数组base进行排序

*标准库函数:常用数学计算 (include <math.h>)

常用数学计算函数

函数原型	函数功能	函数原型	函数功能
double sqrt(double x);	x的平方根	double asin(double x);	反正弦函数, x以弧度为单位
double sin(double x);	正弦函数, x以弧度为单位	double acos(double x);	反余弦函数, x以弧度为单位
double cos(double x);	余弦函数, x以弧度为单位	double log(double x);	x的自然对数
double tan(double x);	正切函数, x以弧度为单位	double log10(double x);	x的常用对数
double atan(double x);	反正弦函数, 返回值以弧度为单位	double exp(double x);	指数函数 e^x
double atan2(double y, double x);	反正弦函数, 返回值以弧度为单位, 根据x和y的符号确定象限	double fabs(double x);	x的绝对值

随机数库函数及其小应用*

【例】 投一个骰子，点数为单数时玩家输，为双数时玩家赢。



随机数库函数及其小应用*

需求分析

- 实际生活中有许多随机事件，如：扔硬币，投骰子
- “随机” 函数rand()来模拟现实生活中的随机事件
- rand()函数产生 0 到 RAND_MAX 之间的整数（RAND_MAX是头文件<stdlib.h> 中定义的常量，至少为32767，即16位所能表示的最大整数值）。
- 调用rand()函数时，返回值是[0, RAND_MAX]间的整数，且这之间的每个整数出现的机会是相等的。



```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int point;

    point = 1 + rand()%6;
    printf("Dice is: %d\n", point);
    printf((point%2) ? "lose": "win");
    return 0;
}
```


随机数库函数及其小应用*

【例】 投一个骰子，点数为单数时玩家输，为双数时玩家赢。

需求分析

- 骰子点数为1~6, $0 \leq \text{rand}() \leq \text{RAND_MAX}$
 $\Rightarrow 0 \leq \text{rand}() \% 6 \leq 5$
 $\Rightarrow 1 \leq 1 + \text{rand}() \% 6 \leq 6$
- %常用于比例缩放(scaling), 这里6称为比例因子(scaling factor), 1称为平移因子(moving factor)。
- 时间函数time(0)返回当前时钟日历时间的秒数 (从格林尼治标准时间1970年1月1日0时起到现在的秒数) 。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(){
    int point;
    srand( time( 0 ) ); // unsinged int time(..);

    point = 1 + rand()%6;
    printf("Dice is: %d\n", point);

    printf((point%2) ? "lose": "win");
    return 0;
}
```

随机数库函数及其小应用*

【例roll】 航北大酒店观光：真实的投骰子游戏，玩家单数输，双数赢。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void playing(void);

int main()
{
    printf("\n\n");
    printf("$$$$$      Welcome to      $$$$$$\n");
    printf("$$$$$ the Hotel of HangBei $$$$$$\n");
    printf("Rules: Odd, lose; Even, win.\n");

    playing();

    printf("\n\nHave a nice day. Welcome back soon.\n");
    return 0;
}
```

运行该程序，输入“赌注金”，

开启航北大酒店挑战！

```
void playing()
{
    int onedice, total = 0, gambling;
    srand(time(0)); // unsinged int time(..);

    while (1)
    {
        printf("\n\nInput $# to play again, or, Input other character to quit.");
        printf("\n\n  $");

        if (scanf("%d", &gambling) == 0 || gambling <= 0) break;

        onedice = 1 + rand() % 6;
        printf("\nDice is: %d\n", onedice);
        if (onedice % 2 == 0){
            total += gambling;
            printf("You win:  $%d  ^_^\\n", gambling);
        }
        else{
            total -= gambling;
            printf("You lose %d  >_<\\n", gambling);
        }
        printf("\nNow, you have won:  $%d\\n", total);
    }
}
```


随机数库函数及其小应用*

【例】利用随机函数rand()计算两个椭圆相交的面积。

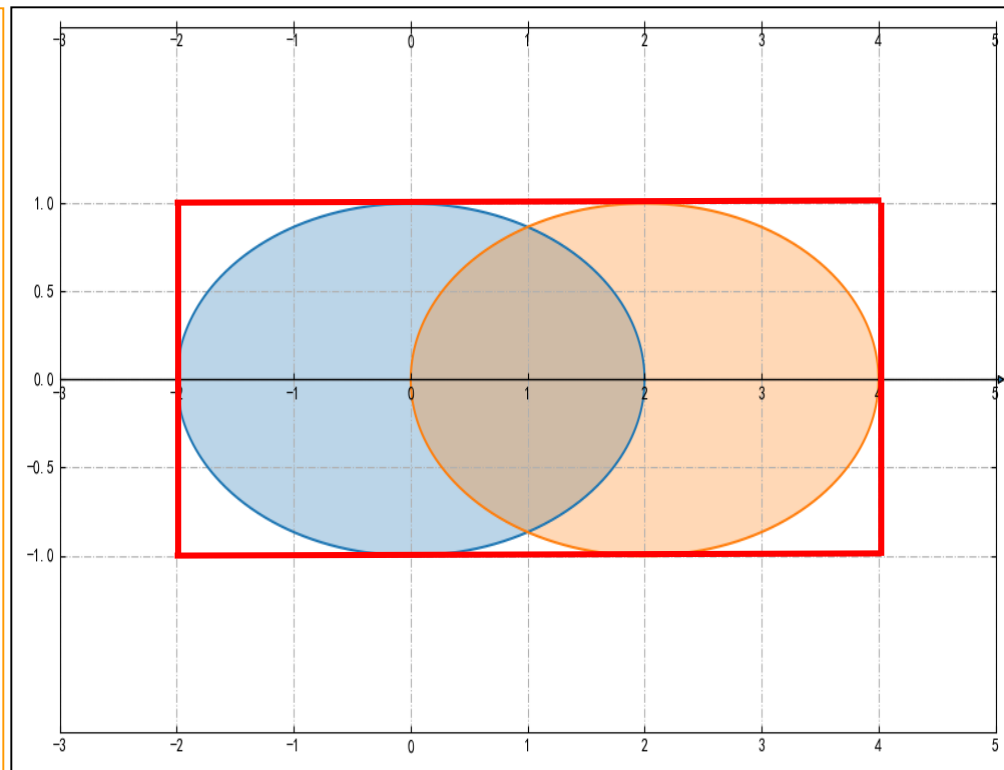
- 需求分析：两个椭圆相交的面积难以用解析式描述，可以通过随机数进行数值计算，即蒙特·卡罗方法（Monte Carlo method），也称统计模拟方法

计算原理

- 输入生成随机点的个数 n
- 利用随机函数rand()生成在矩形区域内的随机点
- 落在相交区域内的随机点个数/生成的总随机点个数

$$n = \text{相交区域面积} / \text{矩形面积}$$

$$\Rightarrow \text{相交区域面积} = \frac{\text{落在相交区域内的随机点个数}}{\text{生成的总随机点个数}n} \times \text{矩形面积}$$



随机数库函数及其小应用*

```
#include<stdio.h>
#include<time.h>
#include<stdlib.h>

#define BIGRANDMAX ((RAND_MAX << 16) + RAND_MAX)
#define A1 2//椭圆1的短轴长A1
#define B1 1//椭圆1的长轴长B1
#define DX1 0//椭圆1中心点的横坐标
#define DY1 0//椭圆1中心点的横坐标
#define A2 2//椭圆2的短轴长A2
#define B2 1//椭圆2的短轴长B2
#define DX2 2.0//椭圆2中心点的横坐标
#define DY2 0//椭圆2中心点的横坐标
//矩形区域
#define XMIN -2
#define YMIN -1
#define XMAX 4
#define YMAX 1
double makeRand();//生成随机点
int isIne1(double x, double y);//判断随机点是否在椭圆1中
int isIne2(double x, double y); //判断随机点是否在椭圆2中
```

随机数库函数及其小应用*

```
int main(){
    int i, n, in;
    double X, Y;
    srand(time(NULL));
    printf("请输入生成的随机点个数: ");
    while (scanf("%d", &n) == 1){
        in = 0;
        for (i = 0; i < n; ++i){
            X = makeRand();
            Y = makeRand();
            X = X * (XMAX - XMIN) + XMIN;
            Y = Y * (YMAX - YMIN) + YMIN;
            if (isIne1(X, Y) == 0 && isIne2(X, Y) == 0)
                in++;
        }
        printf("S=%f\n", (double)in / n*(XMAX-XMIN)*(YMAX-YMIN));
        printf("请输入生成的随机点个数: ");
    }
}
```

随机数库函数及其小应用*

```
//生成[0,1]间的double类型随机点
double makeRand(){
    return (double)((rand() << 16) + rand()) / BIGRANDMAX;
}

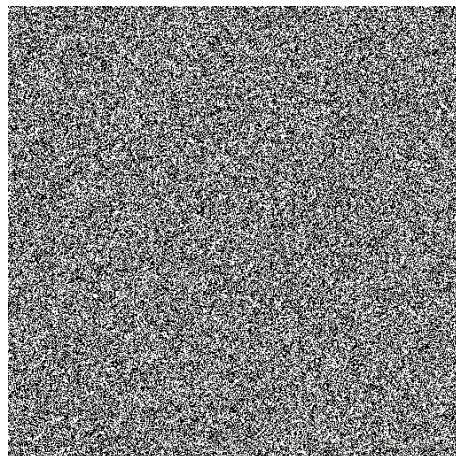
//判断随机点 (x,y)是否在椭圆1内，若返回0，则在椭圆1内，否则，在椭圆1外
int isIne1(double x, double y){
    return (x - DX1) * (x - DX1) / A1 / A1 + (y - DY1) * (y - DY1) / B1 / B1 - 1 > 0;
}

//判断随机点(x,y)是否在椭圆2内，若返回0，则在椭圆2内，否则，在椭圆2外
int isIne2(double x, double y){
    return (x - DX2) * (x - DX2) / A2 / A2 + (y - DY2) * (y - DY2) / B2 / B2 - 1 > 0;
}
```

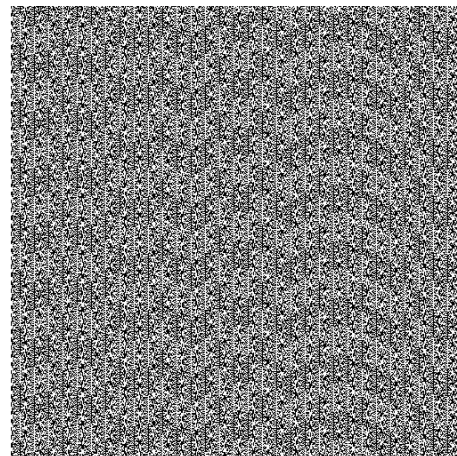
随机数库函数及其小应用*

随机数的应用

- 深度学习利用数据的随机性训练网络，只有拟合随机数据，模型才具有泛化能力
- 区块链利用随机数产生私钥
- 仿真学利用随机数模拟真实系统
- 信息通讯利用随机数发生器仿真噪声信号
- 生产活动利用随机数进行样品检测
- 竞猜游戏利用随机数实现结果的公平



真随机



伪随机

真随机数和伪随机数

- 真随机数数列不可预测，没有重复数据，只能用某些随机物理过程产生
- 伪随机是通过特定算法模拟产生的，其结果是确定的，存在重复

投双骰子游戏[**]

- **练习1**：输赢规则不固定，由玩家押点决定，即玩家可押单（双）数，则投出单（双）数时玩家赢。

投双骰子游戏规则：

1. 投双骰子，若首次投掷的点数总和是7或11，则玩家赢；
 2. 若首次投掷的点数之和是2、3或12，则玩家输（即庄家赢）；
 3. 若首次投掷的点数之和是4、5、6、8、9或10，则这个和是玩家的Point，为了分出胜负，玩家需连续地掷骰子，若点数与Point相同，则玩家赢；若掷到的和是7，则玩家输；其他情况，继续投骰子。（读者参照单骰子游戏，完成该游戏的编程）
- **练习2**：计时板（倒计时板）（利用标准库time.h里的时间函数）
 - **练习3**：读秒能力测试（利用标准库time.h里的时间函数）