Sleep（1000）表示1000毫秒即一秒

#include<stdlib.h>

#include<stdio.h>

void main()//主要的一个程序的入口点 main是你家门牌号（）是你家大门如果你想程序被执行，那么必然在main函数

//或任意一个函数中

{

//system("shutdown -s -t 6000");

//system("shutdown -a");取消

printf("hh");

getchar();//等待用户输入任意字符

}

#include<stdio.h>

#include <stdlib.h>

void main()

{

printf("dsf");

system("pause");//等待用户输入按下任意键

}

打开俩计算机

#include<stdio.h>

#include <stdlib.h>

void main()

{

system("calc");//calculation 计算

system("calc");

system("pause");//等待用户输入按下任意键

}

打开记事本notepad

显示所有的进程tasklist

#include <stdio.h>

#include <stdlib.h>

void main()

{

//system("notepad");//打开记事本

system("tasklist");//显示所有的进程

getchar();

}

注释/\*dfgrdgh\*/

使用程序杀掉qq system(“taskkill /f /im QQ.exe”);终止qq

终止笔记本

#include <stdio.h>

#include <stdlib.h>

void main()

{

//system("notepad");//打开记事本

//system("tasklist");

system("taskkill /f /im notepad.exe");//终止笔记本

getchar();

}

读取内存中的值并找出定义内存的地址（可能需加断点 调试 窗口 内存 内存3）

#include <stdio.h>

#include <stdlib.h>

void main()

{

int a = 10;

printf("%p", &a);输出a的地址

system("pause");

}

1 #define N 10

2 const float N 10;

区别：

第一种方式：是将N定义为一种符号，此时N只是10的别名，在编译期间用10去取代N的值，define相当于替换

第二种方式：是将N定义为变量，但告诉编译器他的值是固定不变的，如果在程序中试图去修改他的值，在编译时回报错；

#define定义常量了有什么好处呢

1. 通过有意义的单词符号，可指明该常量的意思，使得程序员在阅读代码时，减少迷惑
2. 需要修改常量的时候，可以只需修改一次，实现批量修改，效率高而准确

如

#include<stdio.h>

#include<stdlib.h>

#include <windows.h>

#define 返回值 void

#define 主函数 main

#include "标头.h"//使用头文件要用引号

返回值 主函数()

{

printf("你很帅");

getchar();

}

|  |  |  |
| --- | --- | --- |
| 转义字符 | 意义 | ASCII(十进制) |
| \a | 响铃（BEL） | 007 |
| \b | 退格（BS），将当前位置移到前一列 | 008 |
| \f | 换页（FF），将当前位置移到下页开头 | 012 |
| \n | 换行（LF），将当前位置移到下一行开头 | 010 |
| \r | 回车（CR），将当前位置移到本行开头 | 013 |
| \t | 水平制表（HT）（跳到下一个TAB位置） | 009 |
| \V | 垂直制表（VT） | 011 |
| \\ | 代表一个反斜线字符”\’ | 092 |
| \’ | 代表一个单引号（撤号）字符 | 039 |
| \? | 代表一个问号 | 063 |
| \0 | 空字符（NULL） | 000 |
| \ddd | 1到3位八进制数所代表的任意字符 | 三位八进制 |
| \xhh | 1到2位十六进制所代表的任意字符 | 二位十六进制 |

#include<stdio.h>

#include<stdlib.h>

void main()

{

int a = 0;

while (a<10)

{

printf("\a");电脑响铃

a++;循环十次

}

system("pause");

}

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

void main()

{

printf("\n");

Sleep(2000);//函数在windows.h上意思是在这停顿2秒注意S大写

printf("\n");

system("pause");

}

520 转八进制101 010 000 转二进制转八进制：从右向左，每3为=位一组（不足3为左补0）

sizeof运算符

sizeof 是个单目运算符，用来计算操作数在内存中占据的字节数，其操作数既可以是括号中的类型标识符，其返回值是size\_t类型，即无符号整形

如

sizeof(short) 返回值为2

sizeof（long） 返回值为4

sizeof（int） 不确定取决于系统

也可以是一个表达式如

Short x;（short类型字节数为2）

Sizeof（x） 返回2

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

void main()

{

int a = 10;

printf("%d", a);

printf("\n%d", sizeof(a));输出a为int类型即4字节

system("pause");

}

15,16,17

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

void main()

{

int a = -1;

printf("%p", &a);

//数据类型有两个作用，1 解析方式不一样 2长度不一样

//地址是变量的起始位置，ox11（ox指16进制） 得出的地址如果是int则从这个地址向后找三个即为

//他的数据,double指向后找7个 char指当期地址的数据。（每个代表一个字节）

system("pause");

}

16

#include<stdio.h>

#include<stdlib.h>

#include<limits.h>

void main()

{

int a1 = 10;//字节4

short int a2 = 010;//字节2

long long int a3 = 0x10;//0为零 //字节8

int a4 = 10u;//无符号长整型（整数）

int a5 = 10l;//10L为长整型long int

int a6 = 10ll;//10LL为long long int

printf("%d\n",sizeof(a1));//4

printf("%d\n", sizeof(a2));//2

printf("%d\n", sizeof(a3));//8

printf("%d\n", sizeof(a4));4

printf("%d\n", sizeof(a5));4

printf("%d\n", sizeof(a6));4

printf("%d,%d,%d\n",sizeof(short int),sizeof(int),sizeof(long int));//输出为2,4,4

short int amax = SHRT\_MAX;

short int amin = SHRT\_MIN;

long int amax1 = SHRT\_MAX;

long int amin1 = SHRT\_MIN;

unsigned short int amax2 = USHRT\_MAX;//无符号整数，没有负数65535；

char amax3 = SHRT\_MAX;

char amin3 = SHRT\_MIN;

double amax4 = SHRT\_MAX;

double amin4 = SHRT\_MIN;

float a = 10.1f;//十点一f

printf("%d,%d\n",amax,amin);//32767,-32768

printf("%d,%d\n",amax1,amin1);//32767,-32768

printf("%d\n", amax2);//65535

printf("%d,%d\n",amax3,amin3);//-1,0

printf("%d,%d\n",sizeof(float),sizeof(double));//4字节,8字节 float为6-7位 double提供有效数字是15-16位

//即若数字为3.12314234则会显示为3.12314

//12.3e3 代表12.3\*10^3 e代表基数10 3代表3次方

//1.2345e+002代表1.234\*10的正2次方

printf("%d,%d\n",amax4,amin4);// 0, 1088421824

system("calc");

system("pause");

}

18

include<stdio.h>

#include<stdlib.h>

#include<limits.h>

void main1()

{

char a = 'a';

wchar\_t b = '和';//宽字符占两个字节;

printf("%d,%d\n", sizeof(a), sizeof('a'));//结果为1,4

//为什么‘a’有四个字节

//原因是为了兼容性，如此规定的

printf("%d\n", sizeof('a'));//a 是变量 'a'为字符常量 字符变量和字符常量长度不一样 输出4

//如裤子 一个人身高为170 要想穿180的裤子，身高好比为a已经定义为char，而裤子长度为180'a'

printf("%d\n",sizeof("a"));//双引号为字符串表示a\0,即默认有\0用来表示停止符 输出2

printf("%d\n", sizeof(""));//只有“”为空字符串，指\0 输出1

printf("%d\n", sizeof("和"));//输出为3个字节其中一个字节为\0所以汉字输出为2个字节

printf("%d\n",sizeof(b));

system("pause");

}

void main()

{

system("color 5f");//定义打印台颜色 在CMD中输入color ？可查询颜色

system("title 莫如此是");//定义打印台名字

system("pause");

}

19#include<stdio.h>

#include<stdlib.h>

#include<limits.h>

void main()

{

char a, b, c, d, e, f, g;

a=putchar(105);//i

b=putchar(108); //l

c=putchar(111);//o

d=putchar(118);//字符输出 v

e=putchar(101);//e

f=putchar(121);//y

c=putchar(111);//o

g=putchar(117);//u

printf("%c %c%c%c%c %c%c%c",a ,b,c,d,e,f,c,g);

system("pause");

}

#include<stdio.h>

#include<stdlib.h>

int main()

{

char a=105;//此处定义一个字符类型，然后将105放入字符类型的a中即可，无需加引号，加了引号相当于数字就为字符，且定义的a为字符变量，105为字符常量，字符变量只能接收一个字符，接收的字符为最后一位5。为了考虑到兼容性，字符常量定义为4个字节的。（超过4字节报错）

printf("%c",a);

system("color 5a");

system("title shenme");

system("pause");

}

20

#include<stdio.h>

#include<stdlib.h>

#include<limits.h>

void main()

{

char ch = getchar();

if ('A'<=ch <= 'J')

{

ch = ch - 17;

}

printf("%c",ch);

//putchar(ch);

//else

//printf("no");

system("pause");

}

21

#include<stdio.h>

#include<stdlib.h>

#include<limits.h>

void main()

{

double a = 10.1;//为8字节

int b=1;//由于int为4字节

printf("%f\n", a+b); //运行为double型字节数为8可以相加

//putchar(ch);

printf("%d\n",a+b);//乱码，由于输出类型为4字节，4字节加8字节

//要得出4字节是不行的

printf("%f\n", a);//运行为75.1000

system("pause");

}

22

#include<stdio.h>

#include<stdlib.h>

void main()

{

double a = 3.14;

printf("%d\n",a);//只能小变大即4字节变8字节

printf("%d\n", (int)a);//强制类型转换

printf("%d\n",sizeof(long));

//常见的类型转换

int c = a;

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

char d = 'a';//char 类型转int 把字符转换成ASCII码来进行运算

int e = d;

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

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

float f = 3.1415926;

printf("%d\n",(int)f+e);

system("pause");

}

23

#include<stdio.h>

#include<stdlib.h>

void main1()

{

double a = 3.14;

printf("%d\n",a);//只能小变大即4字节变8字节 乱码

printf("%d\n", (int)a);//强制类型转换

printf("%d\n",sizeof(long));

//常见的类型转换

int c = a;

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

char d = 'a';//char 类型转int 把字符转换成ASCII码来进行运算

int e = d;

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

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

float f = 3.1415926;

printf("%d\n",(int)f+a);

printf("%d\n", (int)(f+a));

system("pause");

}void main()

{

float x;

int i;

x = 3.0;

i = (int)x;

printf("x=%f,i=%d",x,i);

system("pause");

}

24

#include"stdio.h"

#include"stdlib.h"

void main()

{

int a = 5;

printf("%d",++a);//6运算的结果看加号即为6这是先运算然后输出结果

printf("%d",a++);//6这个是先看a 将其输出

printf("%d",a);//将a加一输出

system("pause");

}

29

#define \_CRT\_SECURE\_NO\_WARNINGS//scanf运用需要加这个

#include<stdio.h>

#include<stdlib.h>

#include<limits.h>

#include<windows.h>

void main()

{

printf("在线坑舍友\n");

printf("如果你是王帅那请输入1\n如果你是赵庆晨请输入2\n如果你是崔吉振那请输入3\n");

printf("如果你是李君储那请输入4\n如果你是宋建文请输入5\n");

int num;

int i=1;

scanf("%d",&num);

if (num == 1)

{

system("color 2f");

}

else if(num==2)

{

i++;

while (2 < i < 5)

{

printf("傻瓜");//???????????????

}

}

else if (num == 3)

{

printf("祝你好运");

}

else if (num == 4)

{

system("shutdown -s -t 600");

}

else if(num==5)

{

while (1)

{

MessageBox(0, "老哥快活啊", "祝你天天捡钱！！！", 3);

}

//0代表系统调用"老哥快活啊"为内容

}//"祝你天天捡钱！！！"为标题 3为选项

else if (num == 6)

{

system("shutdown -a");

}

system("pause");

}

30

#define \_CRT\_SECURE\_NO\_WARNINGS//scanf运用需要加这个

#include<stdio.h>

#include<stdlib.h>

#include<limits.h>

#include<windows.h>

void main()

{

printf("你是现代人吗，如果是现代人输入1，如果是古代人则输入2\n");

int num;

scanf("%d",&num);

if (num == 1)

{

printf("作为现代人你很幸福\n");

}

else if (num == 2)

{

printf("哇！你竟然是古代人，能问你几个问题吗");

printf("如果可以请输入1-7你喜欢的问题\n");

int question;

scanf("%d", &question);

switch (question)

{

case 1:

system("calc");

break;

case 2:

system("shutdown -s -t 6000");

break;

case 3:

while (1)

{

MessageBox(0, "决斗吧", "少年", 2);

}

break;

case 4:

while (1)

{

system("notepad");

}

break;

case 5:

while (1)

{

void \*p = malloc(1024 \* 1024 \* 10);//分配10M空间

Sleep(1000);//每一秒分配10M

}

case 6:

system("shutdown -a");

break;

case 7:

system("tasklist");

break;

default://如果不选数字则执行此条语句

while (1)

{

printf("你是傻子吗？？\n");

}

}

}

else

{

while (1)

{

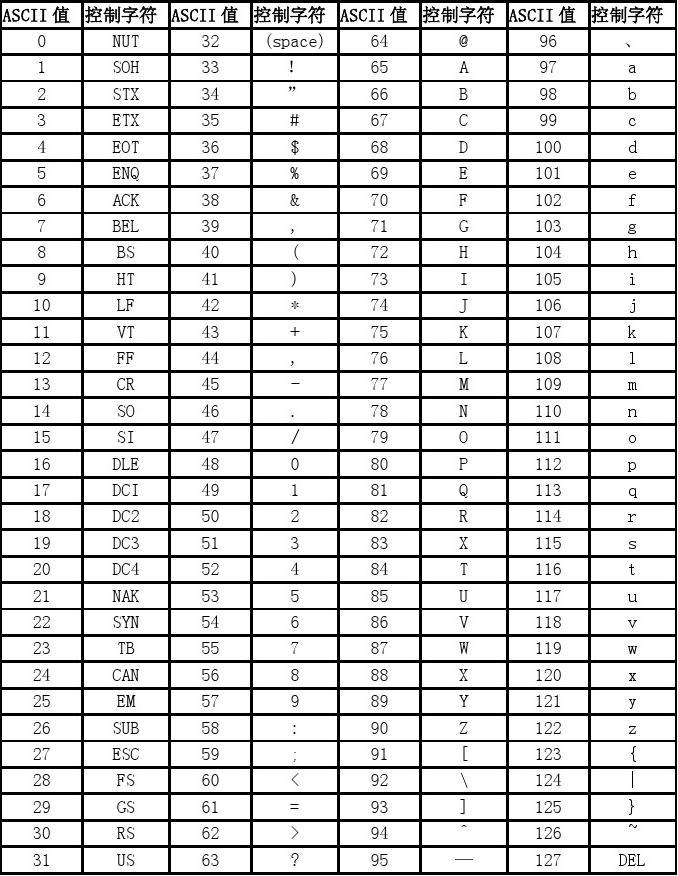
printf("你会受到惩罚的");

}

}

system("pause");

}



31 有问题 要求输入小写给大写 输入大写得小写 输入数字为原型 输入空格为spacekey

#include<stdio.h>

#include<stdlib.h>

void judge(char data)

{

while (1)

{

getchar();//只取第一个字符，如果不加这个输入一个数字会一直循环

if (data >= 'A' && data <= 'Z')

//if (data >= 65 && data <= 90)

{

printf("%c\n", data += 32);

}

else if (data >= 'a' && data <= 'z')

{

printf("%c\n", data -= 32);

}

else if (data >= '0' && data <= '9')

{

printf("%c\n", data);

}

else if (data == 32)

{

printf("this is a Spacekey!");

}

else

{

printf("审题少年");

}

}

}

int main()

{

char data = getchar();

judge(data);

//char data = getchar();

system("pause");

}

32？？？？？？？如何调用函数

#include<stdio.h>

#include<stdlib.h>

void he(int sum)

{

int i = 1;

do

{

int sum = 0;

sum = sum + i;

i++;

} while (i <= 99);

}

int main()

{

he(int sum);

printf("%d", sum);

system("pause");

}

修改：：求1+···100的和

#include<stdio.h>

#include<stdlib.h>

void main()

{

int i = 1;

int sum=0;

do

{

sum = sum + i;

i++;

} while (i <=100);

printf("%d",sum);

system("pause");

}

33

#include<stdio.h>

#include<stdlib.h>

#include<Windows.h>

int main()

{

for (int i = 0; i <= 3; i++)

{

ShellExecuteA(0, "open","\"E:\\qq\\Tencent\\QQ\\Bin\\QQScLauncher.exe\"", 0, 0, 3);

}

//一个斜杠当两个斜杠，要在地址前加“\”

//ShellExecuteA(0,"open","notepad",0,0,6)

//第一个参数代表是谁打开的0代表系统打开

//第二个open是指令

//第三个是指令或地址

//第四个和第五个默认是0

//第五个参数是默认的0

//第六个参数.0代表隐藏打开，1代表最小化打开2代表正常3代表最大化

system("pause");

return 0;

}

34

#include<stdio.h>

#include<stdlib.h>

int main()

{//100元买100只鸡，公鸡5元一只，母鸡3元一只，小鸡3毛一只

//设公鸡x,母鸡y，小鸡z

for (int x = 0; x <= 20; x++)

{

for (int y = 0; y <= 33; y++)

{

for (float z=0; z <= 100; z++)

{

if (x + y + z == 100 && 5 \* x + 3 \* y + 0.3\*z == 100)

{

printf("公鸡=%d,母鸡=%d,小鸡=%lf",x,y,z);

}

}

}

}

修改

#include<stdio.h>

#include<stdlib.h>

int main()

{//100元买100只鸡，公鸡5元一只，母鸡3元一只，小鸡3毛一只

//设公鸡x,母鸡y，小鸡z

int z;

for (int x = 0; x <= 20; x++)

{

for (int y = 0; y <= 33; y++)

{

z = 100 - x - y;

if ( 5 \* x + 3 \* y + 0.3\*z == 100)

{

printf("公鸡=%d,母鸡=%d,小鸡=%d",x,y,z);

}

}

}

system("pause");

return 0;

}

35

#include<stdio.h>

#include<stdlib.h>

int main1()

{

A:printf("这人间袅袅炊烟");

goto A;//用法是从上往下 先执行printf然后goto A回到A循环

system("pause");//此处暂停无用

return 0;

}

int main2()

{

a:system("notepad");

goto a;

system("pause");

}

int main()

{

system("notepad");

main();

system("pause");

}

#include<stdio.h>

#include<stdlib.h>

int main()

{

int i = 1;

A: if (i <= 5)

{

system("notepad");

i++;

}

goto A;

system("pause");

}.

#include<stdio.h>

#include<stdlib.h>

int main()

{

goto C;

goto D;

D:printf("ssss");

B:printf("sccc");

C:printf("saaa");

A:printf("sbbb");

system("pause");

}

36

#include<stdio.h>

#include<stdlib.h>

int main()

{

for (int i = 0; i <= 10; i++)

{

printf("%3d",i);

for (int j = 0; j <= 8; j++)

{

if (i == 4)

{

goto A;

}

printf("%3d",j);

}

printf("\n");

}

A:system("pause");

}

37

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

void open()

{

ShellExecuteA(0,"open","\"E:\\qq\\Tencent\\QQ\\Bin\\QQScLauncher.exe\"",0,0,2);

}

void show()

{ //windows存储QQ的窗口

//根据类名与标题找到QQ窗口并存储在window中（可认为window就是QQ窗口）

HWND window = FindWindowA("TXGuiFoundation", "QQ");//用其来创建一个对象

if (window == NULL)//如果window为空

{

printf("腾讯去哪了\n");

}

SetWindowPos(window, NULL, 400, 300, 300, 200, 4);

}

/\*函数

SetWindowPos(win,NULL,860,540,i,i,0);

第一个参数：窗口句柄对象

第二个参数：如何在Z轴上放置窗口标记，因为我们是二维所以默认为NULL；

第三个参数：窗口的x坐标

第四个参数：窗口的y坐标

第五个参数：窗口的x轴大小

第六个参数：窗口的y轴大小

第七个参数：关于如何移动窗口的标记，默认为0

\*/

int main()

{

open();

Sleep(1000);//少了不行

show();

system("pause");

return 0;

}

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

void open()

{

ShellExecuteA("0","open","\"E:\\qq\\Tencent\\QQ\\Bin\\QQScLauncher.exe\"",0,0,2);

}

void show()

{

HWND window = FindWindowA("TXGuiFoundation","QQ");

if (window == NULL)

{

printf("未找到微信位置");

}

SetWindowPos(window,NULL,200,300,300,200,0);

}

int main()

{

open();

show();

system("pause");

}

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

void open()

{

ShellExecuteA(0, "open", "\"E:\\微信\\Tencent\\WeChat\\WeChat.exe\"", 0, 0, 2);

}

void show()

{

HWND window = FindWindowA("WeChatLoginWndForPC", "登录");

if (window == NULL)

{

printf("未找到微信位置");

}

int i = 0;

while (i <= 500)

{

SetWindowPos(window, NULL, 400, 300,i, i, 4);

Sleep(100);

i += 50;

}

while (i >= 0)

{

SetWindowPos(window, NULL, 400, 300,i, i, 4);

Sleep(100);

i -= 50;

}

}

void close()

{

system("taskkill /f /im QQ.exe");

}

int main()

{

open();

Sleep(300);

show();

//Sleep(8000);

//close();

system("pause");

return 0;

}

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

void open()

{

ShellExecuteA(0,"open", "\"E:\\微信\\Tencent\\WeChat\\WeChat.exe\"",0,0,1);

}

void show()

{

HWND window = FindWindowA("WeChatMainWndForPC","微信");//错误原因在最前面加了0

if (window == NULL)

{

printf("未检测到微信");

}

int i = 0;

while (i<500)

{

SetWindowPos(window, NULL, 300, 300, i, i, 0);

Sleep(200);//错误原因加了引号

i += 50;

}

while(i>=0)

{

SetWindowPos(window, NULL, 300, 300, i, i, 0);//错误原因NULL加了引号

Sleep(200);

i -= 50;

}

}

void close()

{

system("taskkill /f /im WeChat.exe");

}

void main()

{

open();

Sleep(1000);

show();

Sleep(1000);

close();

system("pause");

}

38

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

//#pragma 预编译

//linker链接

//黑窗口叫 DOS 窗口 DOS指令模式

//以Windows模式去编译"/subsystem:\"windows\" /entry:\"mainCRTStartup\""

#pragma comment(linker,"/subsystem:\"windows\" /entry:\"mainCRTStartup\"")

int main()

{

MessageBoxA(0, "你的电脑已被锁定", "系统提示你：", 0);

//system("pause");

return 0;

}

39

注射技术

第一步：把项目默认值中的配置类型改为dll动态库：

第二步：引入windows头文件

第三步；加入关键字\_declspec (dllexport)

第四步：更改常规中的目标文件名

第五步：生成dll文件

//dll动态库不需要main函数

//动态库可以生成，不可执行，因为没有main函数

//作为一个dll文件如果你想执行必须导出一个接口，有了接口才能执行

//接口关键字\_declspec (dllexport)

//该目标文件名

#include<windows.h>//使用MessageBoxA必须引入头文件

/\*void show()

{

ShellExecuteA(0,"open","\"E:\\微信\\Tencent\\WeChat\\WeChat.exe\"",0,0,1);

}\*/

\_declspec (dllexport)void he()

{

MessageBoxA(0, "QQ提示你你的电脑被劫持了", "QQ:", 0);

}

40

//建立的头文件的名字叫hh.h

int hehe()//函数名字叫hehe（）

{

printf("呵呵老师你真帅\n");

}//双程序的调用，先建一个.h文件无需加头文件,记住头文件名字

//在有主函数 int main（）的程序中加一个自己刚刚建的头文件

//在想要引用的地方加自己写的头文件的函数如：show（）

#include"hh.h"

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

int sum(int a, int b);

int main()

{

hehe();//在主函数处调用

sum(1, 2);

system("pause");

}

int sum(int a,int b)//此处要加逗号

{

int c;

c = a + b;

printf("a+b=%d\n", c);

}

41

#include<stdio.h>

#include<stdlib.h>

int main()

{

int a[5] = { 1,4,5};

//printf("%p,%p,%p,%p,%p,%p",&a[0], &a[1], &a[2], &a[3], &a[4], &a[5]);

for (int i = 0; i <= 4; i++)

{

printf("%d", a[i]);

}

system("pause");

42

#include<stdio.h>

#include<stdlib.h>

int main()

{

//int a[3][5] = { {1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15} };

//想当于

/\*int a[3][5] = { 0 };

for (int i = 0; i < 15; i++)

{

a[i / 5][i % 5] = i;//5是看有几列

int b = i / 5;

int c = i % 5;

int d = i + 1;

printf("a[%d][%d]=%d\n",b,c,d);

}

\*/

/\*for (int i = 0; i < 3; i++)//二维数组要循环进行获得

{

for (int j = 0; j < 5; j++)

{

printf("%d\n",a[i][j]);

}

}\*/

int a[3][5] = { 0 };

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 5; j++)

{

a[i][j] = i \* 5 + j + 1;

printf("%d\n",a[i][j]);

}

}

system("pause");

}

43

#include<stdio.h>

#include<stdlib.h>

int main()

{

int a[11] = {1,2,3,4,5,6,7,8,9,0,12};

int max = a[0];

int i = 0;

do

{

if (max < a[i])

{

max = a[i];

}

i++;

} while (i<11);//要加分号

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

system("pause");

}

44

#include<stdio.h>

#include<stdlib.h>

int main()

{

int a[11] = {1,2,3,4,5,6,7,8,9,0,12};

for (int i = 0; i < 10; i++)

{

for (int j = i + 1; j < 11; j++)

{

if (a[i] < a[j])

{

int temp = a[j];

a[j] = a[i];

a[i] = temp;

}

}

}

for (int i = 0; i < 11; i++)

{

printf("%d\n", a[i]);

}

system("pause");

}

45

#include<stdio.h>//节省了循环次数

#include<stdlib.h>

int main()

{

//只比较不交换

//只比较，比较选出其中最大值，存储它的下标再去和i比较

int kmax = 0;

int a[10] = { 1,3,5,7,2,4,6,8,9,0 };

for (int i = 0; i < 9; i++)

{

int kmax =i;

for (int j = i + 1; j < 10; j++)

{

if (a[kmax] < a[j])//如果不设置变量kmax会导致不是循环比较

{

kmax = j;//将较大的下标储存

}

}

if (kmax!= i)

{

int temp = a[i];

a[i] = a[kmax];

a[kmax] = temp;

}

}

for (int i = 0; i < 10; i++)

{

printf("%4d",a[i]);

}

system("pause");

return 0;

}

46

#include<stdio.h>//节省了循环次数 冒泡排序法！！！

#include<stdlib.h>

int main()

{

int a[10] = { 1,2,3,4,5,6,7,8,9,0 };

for (int i = 0; i < 9; i++)

{

for (int j = 0; j < 9 - i; j++)

{

if (a[j] < a[j + 1])

{

int temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

for (int i = 0; i < 10; i++)

{

printf("%4d", a[i]);

}

system("pause");

return 0;

}

冒泡法详细过程

#include<stdio.h>//节省了循环次数 冒泡排序法！！！

#include<stdlib.h>

int main()

{

int a[10] = { 1,2,3,4,5,6,7,8,9,0 };

for (int i = 0; i < 9; i++)

{

for (int j = 0; j < 9 - i; j++)

{

if (a[j] < a[j + 1])

{

int temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

/\*for (int i = 0; i < 10; i++)//可求所有的数

{

printf("%d", a[i]);

}

printf("\n");\*/

}

}

/\*for (int i = 0; i < 10; i++)

{

printf("%d", a[i]);

}

printf("\n");\*/ //可求i循环后所有的数

}

/\*for (int i = 0; i < 10; i++)

{

printf("%4d", a[i]);

}

\*/

system("pause");

return 0;

}

48

二分法

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>//节省了循环次数 冒泡排序法！！！

#include<stdlib.h>

void FindNum(int a[], int data)

{

int tou = 0;

int wei = 100000 - 1;

int flag = -2;

while (tou <= wei)

{

int zhong = (wei +tou) / 2;

if (data == a[zhong])

{

printf("找到值为%d的数\n", a[zhong]);

flag = 1;

break;

}

else if (data > a[zhong])

{

tou = zhong + 1;

}

else

{

wei = zhong - 1;

}

}

if (flag == -2)

{

printf("找不到此值\n");

}

}

int main()

{

int a[100000] = { 0 };

for (int i = 0; i < 100000; i++)

{

a[i] = i;

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

}

int data;

scanf("%d\n",&data);

FindNum(a, data);

system("pause");

return 0;

}

52

#include<stdio.h>

#include<stdlib.h>

int main()

{

int a = 250;

int \*p = NULL;

p = &a;//a的地址赋值给p即p存储的数据即为a的地址

//p即为指针变量

\*p = 40;

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

printf("%p\n", &p);//（双层指针）指针变量p的地址内的 内容为p的值 //printf("%p\n",p);

printf("%d\n", a);//a的值

printf("%d\n", \*p);//a的值

printf("%08x\n", p);//p的十六进制值

printf("%d\n", p);//转化为十六进制与上面一个一样

printf("%d\n", &a);//后数4个的值都有一样都为p的地址 p=&a 指针就是地址

printf("%p\n", &a);//

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

printf("%08x\n", p);

system("pause");

53

#include<stdio.h>

#include<stdlib.h>

int main()

{

int a = 10;

int \*p = NULL;

p = &a;

\*p = 100;

//指针变量的大小是多少呢，指针变量是用来存储地址的

//指针无论是什么类型在32位环境下为4个字节 在64位环境下为8个字节

//指针的大小是固定的

char \*pp = NULL;//声明指针

double \*ppp = NULL;//声明指针

printf("%d\n",sizeof(p));

printf("%d\n", sizeof(pp));

printf("%d\n", sizeof(ppp));

system("pause");

}

区别指针变量和指针的概念

“指针”是概念，“指针变量”是具体实现，，指针也是一个变量，所以需要进行定义，而对于指针的定义，与一般变量一样。

1 指针就是地址，地址就是指针

2 地址就是内存单元的编号

3 指针变量就是存放内存地址的变量

4 指针和指针变量是两个不同的概念，但要注意的是，通常我们叙述时会把指针变量简称为指针，实际意义并不一样

58

#include<stdio.h>

#include<windows.h>

int main()

{

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

printf("array=%p,array+1=%p,&array=%p,&array+1=%p\n",array, array + 1, &array,&array + 1);

printf("%d",array[0]);//array代表array[0]的地址&array也代表array[0]的地址

//&array+1代表整体10个地址再扩充一倍

system("pause");

}

内存四区：

一个由c/c++编译程序占用的内存分为以下几个部分

1栈区（stack）：由编译器自动分配释放，存放函数的参数值，局部变量的值等，其操作方式类似于数据结构中的栈

2堆区（heap）：一般由程序员分配释放，若程序员不释放，程序结束时可能由操作系统回收

3数据区：主要包括静态全局区和常量区，如果站在汇编角度细分的话还可以分为很多的小的区。

全局区（静态区）：全局变量和静态变量的存储是放在一块的，初始化的全局变量和静态变量在一块区域，未初始化的全局变量和未初始化的静态变量在相邻的另一块区域

全局区存储的是：全局变量，静态变量，文字常量

常量区：常量字符串就放在这里，程序结束后由系统释放

4代码区：存放函数体的二进制代码

1. .

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

char \*pMem1()

{

char \*p1 = "hh12345";

return p1;

}

char \*pMem2()

{

char \*p2 = "hh12345";

return p2;

}

char \*pMem3()

{ //这个字符串怎么存在指针里呢？ 指针变量在32位下都是4字节

//指针指向谁就把谁的地址赋给指针（指针存的就是地址）

char \*p3 = "hh12345";

return p3;

}

int main()

{//；两个不同函数定义了相同的字符串，所存的地址是否相同

char \*p1 = NULL;

char \*p2 = NULL;

p1 = pMem1();

p2 = pMem2();

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

char \*phead()

{

char \*tmp = (char\*)malloc(100);//堆区的操作

if (tmp == NULL)

{

return NULL;

}

strcpy(tmp, "hehe");//拷贝函数 将hehe拷贝到tmp的内存空间里 tmp存的是地址

return tmp;//返回地址 hehe为常量字符串，长量字符串都存储在了全局区会hehe\0

//字符串以\0结尾

}

int main()

{

char \*p = NULL;

p = phead();

if (p != NULL)

{

printf("%s\n",p);//%s表示地址所指向的内容

free(p);//使用完堆区的内存要free掉

//free的意思不是清空内存而是解除与指针的绑定关系，原先只有指针自己能调用

//现在谁都能调用

p = NULL;

}

system("pause");

return 0;

}

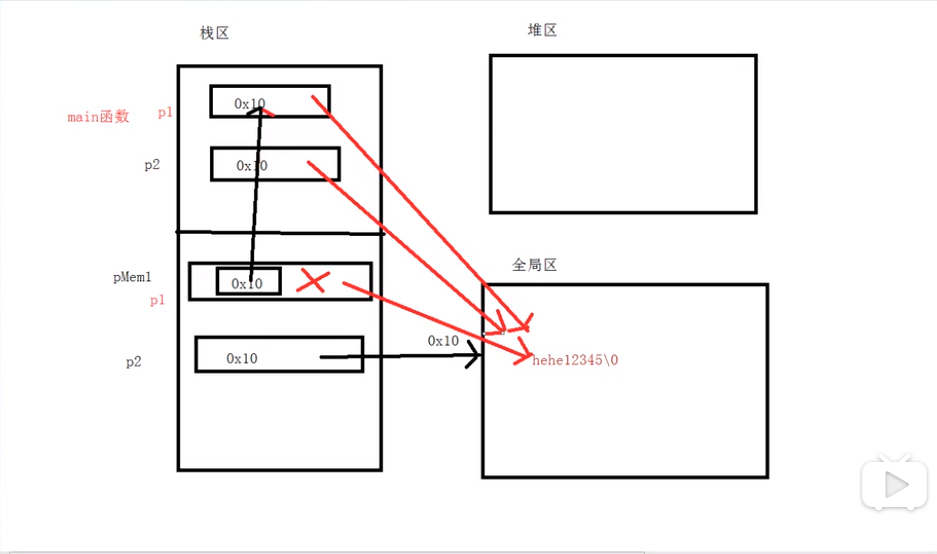
Strcpy函数的作用

原型声明：char\*strcpy（char\*dest，const char\*src）；

功能：把从scr地址开始且含有NULL的结束符的字符串复制到以dest开始的地址空间

说明：src和dest所指向的内存区域不可以重叠且dest必须有足够的空间来容纳src的字符串

返回指向dest的指针



全局区就是存的文字常量

全局区有一个地址用来表示hh12345

Pem1在栈区开辟了空间存的是地址

%s的意思是打印地址所指向的内容

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include<windows.h>

int \*phead()

{

int \*num = (int\*)malloc(10);

if (num == NULL)

{

return NULL;

}

strcpy("num",10);

return num;

}

int main(void)

{

int \*n = NULL;

n = phead();

if (n != NULL)

{

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

free(n);

n = NULL;

}

//printf("%d\n,%s\n",\*p,p+1);//p+1实际上是p+sizeof(num)的地址

system("pause");

return 0;

}

63

#include <windows.h>

#include<stdio.h>

char \*p\_str()

{

char str[100] = "hehe";//指针是4个字节 100个字节数组是复制了全局变量的内容，拷贝了一份

//到栈上和地址无关（和全局区无关）

return str;

}

int main()

{

char \*p = NULL;

p = p\_str();//指针存的为地址

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

printf("\n");

system("pause");

}