第一题

#include <stdio.h>

int main()

{

int a,b; //初始化长和宽

printf("请输入长宽：\n");

scanf\_s("%d%d", &a, &b);

printf("面积为 %d", a \* b); //计算面积

return 0;

}

第二题

#include <stdio.h>

int main()

{

int a, b, c;

printf("请输入三个数，以空格分开\n");

scanf\_s("%d %d %d", &a, &b, &c); //获得三个整数

if (a > b && a > c) //如果a是最大的

{

printf("最大的数=%d", a);

return 0;

}

else

{

if (b > a && b > c) //如果b是最大的

{

printf("最大的数=%d", b);

return 0;

}

else //a、b都不是最大的，那么就是c最大

{

printf("最大的数=%d", c);

return 0;

}

}

return 0;

}

第三题

#include <stdio.h>

int main()

{

int num, a, b, c, Reversenum;

printf("请输入一个三位正整数\n");

scanf\_s("%d", &num);

c = num % 10; // 通过求余获取个位

b = num / 10 % 10; //获得n\*10 后取余得到十位

a = num / 100; //直接/100取整得到百位

Reversenum = c \* 100 + b \* 10 + a; //重新组合为三位数

printf("该三位数逆序数为：%d", Reversenum);

return 0;

}

第四题

#include<stdio.h>

int main()

{

char Char,PreChar,LatChar;

printf("请输入一个字符：\n");

scanf\_s("%c", &Char);

PreChar = Char - 1;

LatChar = Char + 1;

if (PreChar == 96) //如果输入为a

{

PreChar = 122;

}

if (PreChar == 64) //如果输入为A

{

PreChar = 90;

}

if (LatChar == 123) //如果输入为z

{

LatChar = 97;

}

if (LatChar == 91) //如果输入为Z

{

LatChar = 65;

}

printf("%c 的前一个字符为 %c ，后一个字符为 %c", Char, PreChar ,LatChar);

return 0;

}

第五题

#include <stdio.h>

int main()

{

float g, Ct, jin, pound, ounce;//定义 克 克拉 斤 磅 盎司

printf("请输入克数:\n");

scanf\_s("%f",&g);

Ct = g \* 5; //分别计算单位转换

jin = g / 500;

pound = g / 454;

ounce = pound \* 16;

printf("%f 克等价于：\n", g); //输出

printf("%f 克拉\n", Ct);

printf("%f 斤\n", jin);

printf("%f 磅\n", pound);

printf("%f 盎司\n", ounce);

return 0;

}

/\*

对于该程序，我认为浮点更加合适，因为在这里的单位转换会出现非整数的情况。

如果为int则会丢失一定的精度，导致不准

\*/

第六题

#include<stdio.h>

#include<math.h>

int main()

{

signed int SignedInt; //准备所有情况的整数类型

unsigned int UnSignedInt;

signed long SignedLong;

unsigned long UnSignedLong;

signed short SignedShort;

unsigned short UnSignedShort;

char Byte;

//以下依次输出所有整数类型的溢出情况

//SignedINT

printf("\nSignedInt:\n");

SignedInt = pow(2, 31) - 1;

printf("%d \t +1 = %d\n", SignedInt, SignedInt + 1);

SignedInt = -pow(2, 31);

printf("%d \t -1 = %d\n", SignedInt, SignedInt - 1);

//UnSignedINT

printf("\nUnSignedInt:\n");

UnSignedInt = pow(2, 32) -1 ;

printf("%u \t +1 = %u\n", UnSignedInt, UnSignedInt + 1);

UnSignedInt = 0;

printf("%u \t\t -1 = %u\n", UnSignedInt, UnSignedInt - 1);

//SignedLong

printf("\nSignedLong:\n");

SignedLong = pow(2, 31) - 1;

printf("%ld \t +1 = %ld\n", SignedLong, SignedLong + 1);

SignedLong = -pow(2, 31);

printf("%ld \t -1 = %ld\n", SignedLong, SignedLong - 1);

//UnSignedLong

printf("\nUnSignedLong:\n");

UnSignedLong = pow(2, 32) - 1;

printf("%lu \t +1 = %lu\n", UnSignedLong, UnSignedLong + 1);

UnSignedLong = -pow(2, 32);

printf("%lu \t -1 = %lu\n", UnSignedLong, UnSignedLong - 1);

//SignedShort

printf("\nSignedShort:\n");

SignedShort = pow(2, 15) - 1;

printf("%hd \t +1 = %hd\n", SignedShort, SignedShort + 1);

SignedShort = -pow(2, 15);

printf("%hd \t -1 = %hd\n", SignedShort, SignedShort - 1);

//UnSignedShort

printf("\nUnSignedShort:\n");

UnSignedShort = pow(2, 16) - 1;

printf("%hu \t +1 = %hu\n", UnSignedShort, UnSignedShort + 1);

UnSignedShort = 0;

printf("%hu \t -1 = %hu\n", UnSignedShort, UnSignedShort - 1);

//Byte

printf("\nByte:\n");

Byte = pow(2, 8) - 1;

printf("%d \t +1 = %d\n", Byte, Byte + 1);

Byte = 0;

printf("%d \t -1 = %d\n", Byte, Byte - 1);

return 0;

}

第七题

#include <stdio.h>

#include<math.h>

int main()

{

float A, B, C, D; //平面系数

float x, y, z;//点坐标

float distance; //距离

printf("请输入平面系数：\n");

scanf\_s("%f %f %f %f", &A, &B, &C, &D);

printf("请输入点坐标：\n");

scanf\_s( "%f %f %f", &x ,&y, &z);

distance = fabs(A \* x + B \* y + C \* z + D) / sqrt(A \* A + B \* B + C \* C); //计算距离

printf("距离为 %f", distance);

return 0;

}

第八题

#include <stdio.h>

#include <math.h>

struct vector

{

float x;

float y;

float z;

};

float Vcos(vector a, vector b) //计算向量夹角

{

float sum =0;

sum = a.x \* b.x + a.y \* b.y + a.z \* b.z;

sum = sum / sqrt(a.x \* a.x + a.y \* a.y + a.z \* a.z);

sum = sum / sqrt(b.x \* b.x + b.y \* b.y + b.z \* b.z);

return (sum);

}

int main()

{

vector V1, V2;

printf("请输入第一个向量：\n");

scanf\_s("%f %f %f", &V1.x, &V1.y, &V1.z);

printf("请输入第二个向量：\n");

scanf\_s("%f %f %f", &V2.x, &V2.y, &V2.z); //获得两个向量

printf("弧度=%f",acos(Vcos(V1, V2))); //计算夹角并输出

return 0;

}

第九题

#include <stdio.h>

#include <math.h>

bool RunYear(int year) //获取一个年份，闰年结果true，非闰年false

{

if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0)

{

return 1;

}

else

{

return 0;

}

}

int main()

{

int DayInMonth[] = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }; //下标1-12分别为月份的天数

int year, month, day;

int Nowyear, Nowmonth, Nowday;

printf("请输入你的出生日期：\n");

scanf\_s("%d %d %d", &year, &month, &day);

printf("请输入现在日期：\n");

scanf\_s("%d %d %d", &Nowyear, &Nowmonth, &Nowday);

int sum = 0;

for (int i = year + 1; i < Nowyear; i++) { //计算经历的整年一共有多少天

if (RunYear(i) == 1)

{

sum = sum + 366;

}

else

{

sum = sum + 365;

}

}

for (int i = month + 1; i <= 12; i++) { //计算从出生第二个月起到第一年一共多少天

sum = sum + DayInMonth[i];

}

for (int i = 1; i < Nowmonth; i++) { //计算今年从年初到上个月有多少天

sum = sum + DayInMonth[i];

}

if (RunYear(year) && month == 2) { //计算出生日到第一个月末多少天

sum = sum + (29 - day); //如果是闰年二月，要29号开始

}

else

{

sum = sum + (DayInMonth[month] - day);

}

sum = sum + Nowday; //最近一个月的天数

printf("您出生至今一共有 %d 天", sum);

return 0;

}

第十题

#include <stdio.h>

int main()

{

int a,b; //初始化长和宽

printf("请输入长宽：\n");

scanf\_s("%d%d", &a, &b);

printf("面积为 %d", a \* b); //计算面积

return 0;

}

第十一题

#include <stdio.h>

#include<math.h>

int main()

{

float a, b, c, p, area, A, B, C; // abc为三边，ABC为三个角

printf("请输入三边:\n");

scanf\_s("%f %f %f", &a, &b, &c);

p = (a + b + c) / 2; //计算面积并输出

area = sqrt(p \* (p - a) \* (p - b) \* (p - c));

printf("面积=%.4f\n", area);

A = acos((b \* b + c \* c - a \* a) / (2 \* b \* c)); //计算弧度值

B = acos((a \* a + c \* c - b \* b) / (2 \* a \* c));

C = acos((a \* a + b \* b - c \* c) / (2 \* a \* b));

printf("%.2f 对应边弧度为 %f\n", a, A);

printf("%.2f 对应边弧度为 %f\n", b, B);

printf("%.2f 对应边弧度为 %f\n", c, C);

return 0;

}

第十二题

#include <stdio.h>

#include <math.h>

int main()

{

float a, b, c, delta, x1, x2;

printf("请输入二次方程的系数：\n");

scanf\_s("%f %f %f", &a, &b, &c); // 获得二次方程系数

delta = b \* b - 4 \* a \* c;

x1 = ( - b - sqrt(delta))/(2\*a);

x2 = ( - b + sqrt(delta))/(2\*a); //利用求根公式得到两个根

printf("x1 = %f \t x2 = %f \n", x1, x2);

return 0;

}

第十三题

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

int main()

{

//八位二进制数转为十进制

char BitIn[9];

printf("请输入一个八位二进制数：\n");

scanf("%s", BitIn); //获得二进制每一位的ascii

int DecSum = 0;

for(int i =0;i <= 7;i++) //按照不同位的系数相加

{

DecSum = DecSum + (BitIn[i]-48) \* pow(2, (7 - i)); // asc-48得到该位的值，2的7-i次方得到系数

}

printf("二进制对应十进制为 %d\n", DecSum);

//将十进制数转为二进制数

long DecIn;

long long BitOut = 0;

printf("请输入一个十进制数：\n");

scanf\_s("%d", &DecIn); //获得十位数

int i = -1; // 用于记录是第几次短除

long remaind; //用于记录余数

while (DecIn >0) //只要剩下的不为0就继续短除

{

i++;

remaind = DecIn % 2; //取得余数

BitOut = BitOut + remaind \* pow(10,i); //每一次的余数位都高一级，为了在long中输出，直接乘以10变成下一位

DecIn = (DecIn - remaind) / 2; //剩下的数进行下一轮短除

}

printf("该十进制数对应二进制为 %lld\n", BitOut);

return 0;

}

第十四题

#include <stdio.h>

int main()

{

printf("△\n\n"); //显然不是这样

int size;

printf("请输入目标大小：\n");

scanf\_s("%d", &size);

printf("\n\n");

if(size ==1)

{

printf("△\n"); //最小三角形

}

else

{

int Line;

Line = size - 2; //三角形对应边字符数量

for (int i = 1; i <= size;i++)

{

printf("%d\t", i); //输出数轴

if (i == 1) //处理第一行的星号

{

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

{

printf(" ");

}

printf("\*");

}

else if(i == size) //处理最后一行的星号与连接号

{

printf("\*");

for (int j = 1; j <= 2 \* Line + 1; j++)

{

printf("-");

}

printf("\*");

}

else //处理中间几行的缩进

{

for (int j = 0; j <= Line +1 - i; j++)

{

printf(" ");

}

printf("/");

for (int j = 1; j <= 2\*i-3; j++)

{

printf(" ");

}

printf("\\");

}

printf("\n");

}

}

return 0;

}