第一题

#include <stdio.h>

int main()

{

int a = 3, b = 4, c = 5;

double x = 1.2, y = 2.4, z = -3.6;

long u = 51274, n = 128765;

char c1 = 'a', c2 = 'b';

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

printf("x = %.6lf ; y = %.6lf ; z = %.1lf ;\n", x, y, z);

printf("x + y = %.2lf ; y + z = %.2lf \n", x + y, y+ z);

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

printf("c1 = %d; \n",c1);

printf("c2 = %d; \n", c2);

return 0;

}

第二题

第六行：a=5 b=2

第七行: a=5 b=5

第八行：a=5 b=5

第三题

第六行：x=10 y=5

第七行: x=10 y=15

第八行: x=150 y=15

第四题

#include <stdio.h>

int main()

{

printf("Lu TianXing\n");

printf("Lu\nTianXing\n");

printf("Lu");

printf(" TianXing\n");

return 0;

}

第五题

#include <stdio.h>

int main(void)

{

int i = 1;

float n;

printf("Watch out! Here come a bunch of fractions!\n");

for(;i<=30;i++)

{

n = 1.0/i;

printf("%f ", n);

}

printf("That's all, folks!\n");

return 0;

}

第六题

SOS:4 4.00

第七题

#include <stdio.h>

#include <math.h>

int main()

{

int a = 10 ,b = 2;

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

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

printf("a^3 = %d \n", a\*a\*a);

printf("b^3 = %d \n", b\*b\*b);

printf("a^3 / b^3 = %d \n", (a \* a \* a)/(b \* b \* b));

//修改后

double c = 10, d = 2;

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

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

printf("a^3 = %llf \n", pow(c,3));

printf("b^3 = %llf \n", pow(d,3));

printf("a^3 / b^3 = %llf \n", pow(c, 3) / pow(d, 3));

return 0;

}

/\*

不合理的原因：整形除以整形，默认还是整形，导致小数点后被舍弃

修改方式，将整形转为浮点

\*/

第八题

#include<stdio.h>

int main()

{

char c1, c2;

c1 = getchar();

c2 = getchar();

putchar(c1);

putchar(c2);

printf("\n%c", c1);

printf("%c", c2);

return 0;

}

/\*

c1/c2的定义应该为字符型整形均可

要求输出ASCII应该使用printf

整形和字符在char取值内时可以直接转，当超出时，存在问题

\*/

第九题

%s! C is cool!

! C is cool!

11

11

12

11

第十题

#include <iostream>

int main()

{

printf("abc\b\b\bdef"); //def 存在有三个退格符号(\b)，将输出的abc删掉

printf("abc\rdef"); /\* def 先打印出 abc 中间存在回车(\r), 重新从该行输入，用def覆盖abc\*/

printf("abc\ndef"); // abc [换行]def \n换行

printf("abc\tdef \n"); //abc def 中间有\t 作为tab缩进

printf("He said：\" I don't know \" \n");

printf("\141 \142 \143 \n");

printf("\x61 \x62 \x63 \n");

}

第一题

#include <stdio.h>

#include <math.h>

int main()

{

float x, y; //定义x与f(x)

printf("请输入x的值：\n");

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

if (x < 0) //分段计算

{

y = 4 \* x - 3;

}

else

{

y = pow((x - 2), 2) - 7;

}

printf("f(x)为%f", y);

return 0;

}

第二题

#include <stdio.h>

int GuessNum[] = { 6,1,8 };

int main()

{

int TryNum, SplitNum[3]; //分别定义输入数，输入数各位

int NumRight = 0, PositionRight = 0; //正确的数，正确位置的数

printf("请输入猜的数：\n");

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

SplitNum[0] = TryNum / 100; //获取输入数的各个位

SplitNum[1] = TryNum / 10 % 10;

SplitNum[2] = TryNum % 10;

for (int i = 0; i <= 2;i++)

{

for (int j = 0; j <= 2;j++)

{

if (GuessNum[i] == SplitNum[j]) //相互遍历一遍，进行匹配

{

NumRight++; //如果数是正确的

if (i == j)

{

PositionRight++; //如果位置是正确的

}

}

}

}

printf("%d 个数正确，%d 个数位置正确",NumRight,PositionRight);

return 0;

}

第三题

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main()

{

double Num1 = 0, Num2 = 0, Result; // 定义第一个第二个数，和结果

char Operator; //定义操作符

printf("请输入两个数和一个操作符：\n");

scanf("%lf %c %lf", &Num1,&Operator ,&Num2);

//printf("%f %f %d \n", Num1, Num2, Operator);

switch ((int)Operator) //根据操作符判断

{

case 43: Result = Num1 + Num2; break; //+

case 45: Result = Num1 - Num2; break; //-

case 42: Result = Num1 \* Num2; break; //\*

case 47: // /

if (Num2 != 0)

{

Result = Num1 / Num2; break;

}

else

{

printf("ERROR");return 0;

}

case 37: // %

if (Num2 != 0)

{

Result = (int)Num1 % (int)Num2; break;

}

else

{

printf("ERROR");return 0;

}

default: printf("非法运算符");return 0;

}

printf("%.2lf %c %.2lf = %.2lf", Num1, Operator, Num2, Result);

return 0;

}

第四题

#include <stdio.h>

int main()

{

int StudentNum = 0, TotalScore = 0, InputScore, FailedNum = 0; //学生总数，总分，输入分数，不及格人数

float AvgScore,FailRate; //平局分 不及格率

printf("请输入学生成绩:\n");

for (;;) //死循环连续输入

{

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

if (InputScore >= 0 && InputScore <= 100) //如果分数合法

{

StudentNum++; //学生数+1，总分+输入分数

TotalScore = TotalScore + InputScore;

if (InputScore < 60)

{

FailedNum++; //不及格时，不及格人数加一

}

}

else

{

break; //输入的数不合法，则跳出循环

}

}

AvgScore = (float)TotalScore / StudentNum;

FailRate = (float)FailedNum / StudentNum;

printf("平均成绩：%f \n 总人数：%d \n 不及格率: %f \n", AvgScore, StudentNum, FailRate);

return 0;

}

第五题

#include <stdio.h>

#include <math.h>

int main()

{

double m, PayOut = 0, MakeIn = 0; //m大小，富豪支出，富豪收入

printf("请输入m\n");

scanf\_s("%lf", &m);

for (int i = 1; i <= 30; i++) //计算每天支付的量，得到总额

{

PayOut = PayOut + 0.01 \* pow(2, i - 1);

}

MakeIn = m \* 30; //计算总收入

printf("亏损为：%f", PayOut - MakeIn);

return 0;

}

第六题

#include <stdio.h>

int main()

{

int Num;

printf("下面的数字不能说：\n");

for(int Num\_100 = 0; Num\_100 <=2; Num\_100++) // 分别遍历 百位 十位 个位 的所有可能

{

for (int Num\_10 = 0; Num\_10 <= 9; Num\_10++)

{

for (int Num\_1=0; Num\_1 <= 9; Num\_1++)

{

Num = Num\_100 \* 100 + Num\_10 \* 10 + Num\_1;

if (Num > 200)

{

return 0;

}

if ( Num % 7 == 0 || Num\_100 == 7 || Num\_10 == 7 || Num\_1 ==7) //两百以内，且符合含有7的要求

{

printf("%d,",Num);

}

}

}

}

return 0;

}

第七题

#include <stdio.h>

int main()

{

int Price = 64, GuessPrice;

printf("请输入商品价格(最多五次)：\n");

for (int i = 1; i <= 5; i++) //最多输入5次

{

scanf\_s("%d", &GuessPrice); //获取猜测

if (GuessPrice == Price) //分类讨论价格

{

printf("You Get It!"); //猜测正确输出并退出

return 0;

}

else if (GuessPrice > Price)

{

printf("你猜的价格高了\n");

}

else

{

printf("你猜的价格低了\n");

}

}

printf("Sorry!"); //五次全部失败退出

return 0;

}

第八题

#include <stdio.h>

void ReSort(int arrInt[],int size,bool BiggerFirst)//冒泡排序，size为sizeof(arr)/sizeof(arr[0])

{

int temp;

for (int i = 0; i <= size - 2; i++)

{

for (int j = 0; j <= size - 2 - i; j++)

{

if (BiggerFirst)

{

if (arrInt[j] < arrInt[j + 1]) //从大到小排列

{

temp = arrInt[j + 1];

arrInt[j + 1] = arrInt[j];

arrInt[j] = temp;

}

}

else

{

if (arrInt[j] > arrInt[j + 1]) //从小到大排列

{

temp = arrInt[j + 1];

arrInt[j + 1] = arrInt[j];

arrInt[j] = temp;

}

}

}

}

}

int Restruct(int NumIn) //重构数字，生成最大最小两个数字，返回差

{

int Num[4],Max,Min;

Num[0] = NumIn / 1000; //获取各个位上的数字

Num[1] = NumIn / 100 % 10;

Num[2] = NumIn / 10 % 10;

Num[3] = NumIn % 10;

ReSort(Num, 4,true); //重新排序

Max = Num[0] \* 1000 + Num[1] \* 100 + Num[2] \* 10 + Num[3];

Min = Num[3] \* 1000 + Num[2] \* 100 + Num[1] \* 10 + Num[0];

return Max - Min;

}

int main()

{

int LastResult,step = 0;

printf("请输入一个四位正整数:\n");

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

while (LastResult != 6174) //迭代重组数，直到为6174后输出步骤

{

step++;

LastResult = Restruct(LastResult);

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

}

printf("%d", step);

return 0;

}

第九题

#include <stdio.h>

#include <math.h>

int main()

{

float a,Result,LastResult; //输入的a，记录Xn，Xn+1

printf("请输入a：\n");

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

LastResult = a / 2;

Result = (LastResult + a / LastResult) / 2; //先进行猜测a/2，后进行比较

while (fabs(LastResult - Result) > pow(10, -5)) //如果 Xn+1 - Xn不符合就继续循环

{

LastResult = Result; //记录Xn，并生成新的结果

Result = (LastResult + a / LastResult) / 2;

}

printf("%f",Result);

return 0;

}

第十题

#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[25565];

printf("请输入一个二进制数：\n");

scanf("%s", BitIn); //获得二进制每一位的ascii

int lens;

for (lens = 0; BitIn[lens] != '\0'; lens++); //得到输入的二进制长度，lens包含最后的\0结尾，实际长度lens - 1

int DecSum = 0;

for (int i = 0;i <= lens - 1;i++) //按照不同位的系数相加

{

DecSum = DecSum + (BitIn[i] - 48) \* pow(2, (lens -1 - i)); // asc-48得到该位的值，2的lens - 1 - 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 long DecIn;

long long BitOut = 0;

printf("请输入一个十进制数：\n");

scanf\_s("%lld", &DecIn); //获得十位数

long long i = -1; // 用于记录是第几次短除

long 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;

}

第十一题

#include <stdio.h>

#define Size 10

int main()

{

//正向三角

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

{

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

{

printf(" ");

}

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

{

printf("\*");

}

printf("\n");

}

//逆向三角

printf("\n");

for (int i = Size; i >= 1; i--)

{

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

{

printf(" ");

}

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

{

printf("\*");

}

printf("\n");

}

//半三角

printf("\n");

for (int i = Size / 2; i <= Size; i++)

{

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

{

printf(" ");

}

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

{

printf("\*");

}

printf("\n");

}

return 0;

}

第十二题

#include <stdio.h>

int main()

{

int n;

printf("请输入n:\n");

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

for (int i = 2; i <= n;i++) //从2循环到n

{

bool Prime = true; //默认判断它是质数

for (int j = 2; j <= i / 2 + 1; j++) //从2到n/2+1循环，尝试是否能整除

{

if ( i % j == 0 && i !=j) //如果可以整除且除数不等于它本身，判断它不是质数

{

Prime = false;

break;

}

}

if (Prime == true) //根据判断输出对应的文字

{

printf("%d 是质数\n", i);

}

else

{

printf("%d 不是质数\n", i);

}

}

return 0;

}

第十三题

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

int main()

{

int len; //文本长度

char StrIn[25565];

printf("请输入字符串:\n");

scanf("%s", StrIn);

len = strlen(StrIn); //获得文本长度

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

{

int j = len - 1 - i; //i从0开始，j从len-1开始，向中间遍历

if (i >= j) //如果 i 超过j 说明判断完成，没有非对称文本

{

printf("是回文");

break;

}

if (StrIn[i] != StrIn[j]) //发现非对称文本，输出不是回文

{

printf("非回文");

break;

}

}

return 0;

}