第一题

代码：

#include <iostream>

using namespace std;

int main()

{

int n, a[100], b[100], c;

while (1)

{

cout << endl << "请输入N的值:";

cin >> n;

c = n;//保留n原始值

for (; n > 0; n--)

{

for (int i = n - 1; i >= 0; i--)//加数赋值

b[i] = 2;

for (int i = n - 1; i >= 0; i--)

{

a[i] += b[i];//逐位相加

while (a[i] > 9)//while确保某一位小于10

{

a[i] -= 10;

a[i + 1]++;//进位

}

}

}

cout << "精确计算的值为 : ";

for (int i = c - 1; i >= 0; i--)//输出结果

{

cout << a[i];

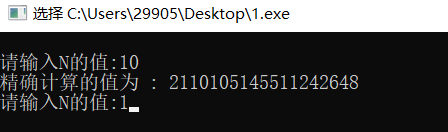
a[i] = 0;//用于继续测试，数组回到初始状态

}

}

}

运行截图：



第二题

代码：

#include <iostream>

using namespace std;

int main()

{

int a[20], n;

while (1)

{

cout << "请输入要输入的整数数量(<20) : ";

cin >> n;

cout << "请输入" << n << "个数字（逐一输入或空格间断） : " << endl;//cin的特性：空格间断

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

{

cout << "No ." << i + 1 << " :";

cin >> a[i];

}

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

{

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

{

cout << a[i] << " ";//输出数组

}

cout << endl;

int temp = a[n - 1];//取出末尾的值

for (int i = n - 1; i >= 0; i--)//循环从后往前，给后面的数赋前面的值

{

if (i != 0)//当i=0时直接给末尾赋值

a[i] = a[i - 1];

else

a[i] = temp;

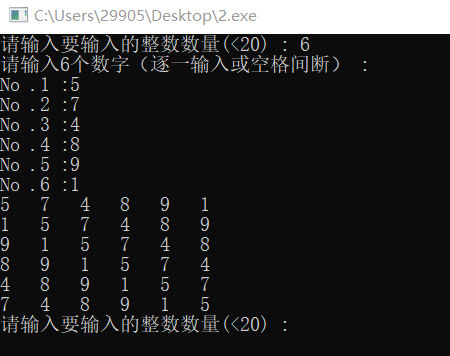
}

}

}

}

运行截图：



第三题

代码：

#include <iostream>

using namespace std;

int main()

{

int i, j, n, k = 1, a[20][20];

while (1)

{

cout << "请输入矩阵的边长 : ";

cin >> n;

for (i = 0; i < n / 2; i++)//计算出环的总数

{

for (j = i; j < n - i - 1; j++)

a[i][j] = k++;//矩阵第一部分：上方

for (j = i; j < n - i - 1; j++)

a[j][n - i - 1] = k++;//矩阵第二部分：右边

for (j = n - i - 1; j > i; j--)

a[n - i - 1][j] = k++;//矩阵第三部分：下方

for (j = n - i - 1; j > i; j--)

a[j][i] = k++;//矩阵第四部分：左边

}

if (n % 2 != 0)//当边长为奇数时，给正中间单独的一个数赋值

a[(n - 1) / 2][(n - 1) / 2] = n \* n;

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

cout << a[i][j] << "\t";//输出结果

}

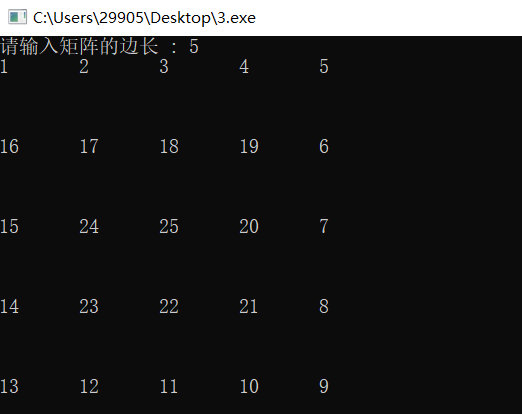
cout << endl << endl << endl << endl;

}

}

}

运行截图：



第四题

代码：

#include <iostream>

using namespace std;

int main()

{

int i, n, a[20][20], k, temp2 = 0, x, y;

while (1)

{

for (x = 0; x <= 19; x++)

for (y = 0; y <= 19; y++)

a[x][y] = 0;//数组所有值初始化为0

k = 1;//k值重置，用于再次输出三角形

cout << "请输入上三角形上方的边长 : ";

cin >> n;

for (i = 0; i < n; i++)

{

a[i][0] = k++;//环的第一个数据

for (int temp1 = i; temp1 > 0;)//temp1的操作在for循环中会导致temp1的值过早或过晚的改变

{

a[--temp1][++temp2] = k++;//所以把temp1的递减操作放到了循环里面

}

temp2 = 0;//每次使用完temp2后重置temp2的值用于下个环使用

}

for (x = 0; x < n; x++)

{

for (y = 0; y < n; y++)

{

if (a[x][y] != 0)//初始化所有值默认为0，而正常运行时不会出现0，所以排除0的数据

cout << a[x][y] << "\t";

}

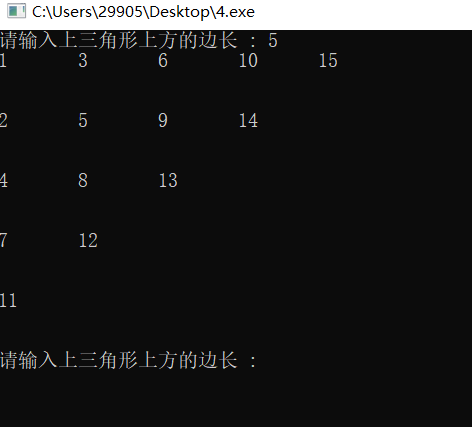
cout << endl << endl << endl;

}

}

}

运行截图：



第五题

代码：

#include <iostream>

using namespace std;

int main()

{

int i, j, r, n, a[50][50];//C++的{}初始化更简便

while (1)

{

cout << "请输入矩阵的边长 : ";

cin >> n;

r = n / 2;//求出环数（不包括只有一个数字的环）

for (i = 0; i < r; i++)

{

for (j = i; j < n - i - 1; j++)//上方

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

for (j = i; j < n - i - 1; j++)//右边

a[j][n - i - 1] = i + 1;

for (j = n - i - 1; j > i; j--)//下方

a[n - i - 1][j] = i + 1;

for (j = n - i - 1; j > i; j--)//左边

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

}

if (n % 2 != 0)//矩阵边长为奇数时触发

a[(n + 1) / 2 - 1][(n + 1) / 2 - 1] = r + 1;//中心点赋值

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

{

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

{

cout << a[x][y] << "\t";//输出结果

}

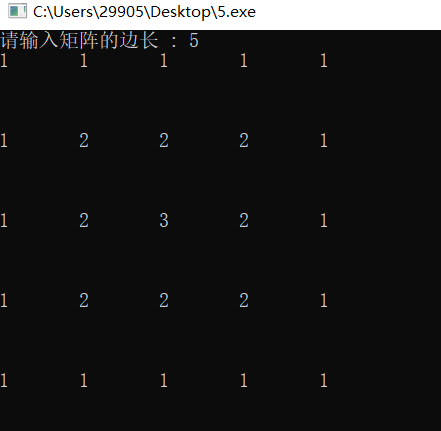
cout << endl << endl << endl << endl;

}

}

}

运行截图：



第六题

代码：

#include<iostream>

#include<string>

using namespace std;

string s;

int main()

{

int kh[100], L[10], R[10], i, j, min, max, tempmin = 100, x = -1, y = -1;

while (1)

{

cout << "请输入一行四则运算表达式 : " << endl;

getline(cin, s);

for (i = 0; i < s.length(); i++)//根据String长度遍历

{

if (s[i] == '(')//左括号则赋值为1

kh[i] = 1;

if (s[i] == ')')//右括号则赋值为2

kh[i] = 2;

}

for (i = 0; i < s.length(); i++)//再次遍历

{

if (kh[i] == 1)

{

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

{

if (L[temp] == 0)//数组temp索引等于0则存入左括号坐标

{

L[temp] = i;

break;

}

else

continue;//不等于，即已经存入过了数据，则继续使索引加一再次判断

}

}

if (kh[i] == 2)

{

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

{

if (R[temp] == 0)//数组temp索引等于0则存入右括号坐标

{

R[temp] = i;

break;

}

else

continue;//不等于，即已经存入过了数据，则继续使索引加一再次判断

}

}

}

for (int temp = 0; temp < 5; temp++)//最多支持五个括号对同时出现，可通过修改最大值提升

{

tempmin = 100;

x = -1;

y = -1;

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

{

for (j = 0; j < 9; j++)

{

if (R[j] - L[i] > 0 && R[j] - L[i] < tempmin)

{

tempmin = R[j] - L[i];

max = L[i];//max赋值差距最小的左括号坐标

min = R[j];//min赋值差距最小的右括号坐标

x = j;//记录最小的右括号坐标

y = i;//记录最小的左括号坐标

}

}

}

if (x != -1 && y != -1)//已经使用过的数据归零，避免再次调用

{

R[x] = 0;

L[y] = 0;

}

//下一行代码用于输出最下差值的坐标

//cout << "max : " << max << " min : " << min << endl;

if (max == 0 && min == 100)

break;

if (max == 0 && min == s.length() - 1)//左括号坐标为0右括号为最右边的情况

{//此情况不用考虑左边为减号的情况

s[max] = ' ';

s[min] = ' ';//修改原输入的string的值

}

else if (max != 0 && min != s.length() - 1)//左括号坐标不为0右括号不在最右边的情况

{

if (s[max - 1] == '\*' || s[max - 1] == '/' || s[min + 1] == '\*' || s[min + 1] == '/')//判断是否为不可去掉括号的符号

continue;

else if (s[max - 1] == '-')//考虑左边为减号的情况，修改括号内的加减号

{

for (i = max; i < min; i++)

{

if (s[i] == '+')

{

s[i] = '-';

s[max] = ' ';

s[min] = ' ';

break;//退出循环，否则会多次赋值

}

if (s[i] == '-')

{

s[i] = '+';

s[max] = ' ';

s[min] = ' ';

break;//退出循环，否则会多次赋值

}

}

}

else

{

s[max] = ' ';

s[min] = ' ';//修改原输入的string的值

}

}

else if (max == 0 && min != s.length() - 1)//左括号坐标为0右括号不在最右边的情况

{

if (s[min + 1] == '\*' || s[min + 1] == '/')//判断是否为不可去掉括号的符号

continue;

else

{

s[max] = ' ';

s[min] = ' ';//修改原输入的string的值

}

}

else if (max != 0 && min == s.length() - 1)//左括号坐标不为0右括号在最右边的情况

{

if (s[max - 1] == '\*' || s[max - 1] == '/')//判断是否为不可去掉括号的符号

continue;

else if (s[max - 1] == '-')

{

for (i = max; i < min; i++)//考虑左边为减号的情况，修改括号内的加减号

{

if (s[i] == '+')//左括号的左边是减号的情况下，括号内的加号改为减号

{

s[i] = '-';

s[max] = ' ';

s[min] = ' ';

break;//退出循环，否则会多次赋值

}

if (s[i] == '-')//左括号的左边是减号的情况下，括号内的减号改为加号

{

s[i] = '+';

s[max] = ' ';

s[min] = ' ';

break;//退出循环，否则会多次赋值

}

}

}

else

{

s[max] = ' ';

s[min] = ' ';//修改原输入的string的值

}

}

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

{//下面这一行代码用于测试时输出左右括号的坐标

//cout << "L[" << i << "] : " << L[i] << " R[" << i << "] : " << R[i] << endl;

if (L[i] == max)

L[i] = 0;

if (R[i] == min)

R[i] = 0;

}

}

cout << "输出结果 : ";

for (i = 0; i < s.length(); i++)//输出时跳过空格

{

if (s[i] != ' ')

cout << s[i];

}

cout << endl;

max = 0;

min = 100;

for (i = 0; i < 9; i++)//坐标记录数组复原

{

L[i] = 0;

R[i] = 0;

}

for (i = 0; i < 99; i++)//标记坐标数组复原

{

kh[i] = 0;

}

}

}

运行截图：



第七题

代码：

#include <iostream>

using namespace std;

int ack(int m, int n)

{

if (m == 0)

return n + 1;

else if (n == 0)

return ack(m - 1, 1);

else

return ack(m - 1, ack(m, n - 1));

}

int main()

{

int m, n;

cout << "输入M、N的值（空格间断）:";

cin >> m >> n;

cout << ack(m, n);

}

运行截图：



第八题

代码：

#include <iostream>

#include<string>

using namespace std;

string s;

bool Check(int L,int R)

{

if (L == R)//字符串长度为单数时，判断到左右索引相等为止

return true;

if (R - L == 1 && s[L] == s[R])//字符串长度为偶数时，判断到左右索引相减为一为止

return true;

if (s[L] == s[R])//左右索引指向的字符相等时继续递归执行

return(Check(L + 1, R - 1));

else//如果不相等，则返回false

return false;

}

int main()

{

while (1)

{

cout << "请输入一个字符串 : ";

cin >> s;

if (Check(0,s.length()-1))//布尔值

cout << "该字符串为回文" << endl;

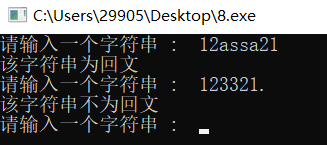
else

cout << "该字符串不为回文" << endl;

}

}

运行截图：



第九题

代码：

#include <iostream>

using namespace std;

int t[100];

int Check(int c)

{

if (c < 0)//b减a小于0时，由于不能向右走，所以可行路线为0

return 0;

else if (c == 0 || c == 1)//b减a为0或1时，到自己或自己加一的可行路线为1

return 1;

else

return(Check(c - 1) + Check(c - 2));//都不满足就返回c减1和c减2，继续递归计算

}

int main()

{

int a, b;

while (1)

{

cout << "请输入起始位置-蜂房a : ";

cin >> a;

cout << "请输入终止位置-蜂房b : ";

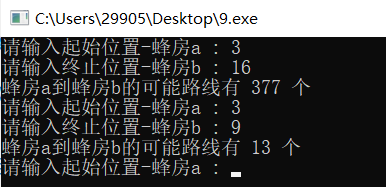
cin >> b;

cout << "蜂房a到蜂房b的可能路线有 " << Check(b - a) << " 个" << endl;

}

}

运行截图：



第十题

代码：

#include <iostream>

using namespace std;

int main()

{

int n, m, cave[20], flag;

while (1)

{

cout << "请输入洞的数量 : ";

cin >> n;

cout << "请输入狼每经过几个洞就进洞 : ";

cin >> m;

for (int x = 2; x < m; x++)//检测m和n的最小公倍数，从2开始可以不判断公倍数为1的情况

{

for (int y = 2; y < n; y++)

{

if (n \* x == m \* y && (x != m && y != n))//搜寻最小公倍数

flag = 1;

}

}

if (n % m == 0 || flag)

cout << "兔子有机会生存!" << endl;

else//在n不是m的非零整数倍且无非一公倍数的情况下，兔子没有机会生存

cout << "兔子无法幸存" << endl;

for (int i = 0; i < n\*n; i += m)

cave[i%n] = 1;//i%n在i值超出n时，变到下一个周期的合理位置

if (n % m == 0 || flag)

{

cout << "兔子躲在以下位置可以生存(从0开始) :";

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

{

if (cave[i] != 1)//输出狼没进过的洞的编号

cout << i << " ";

}

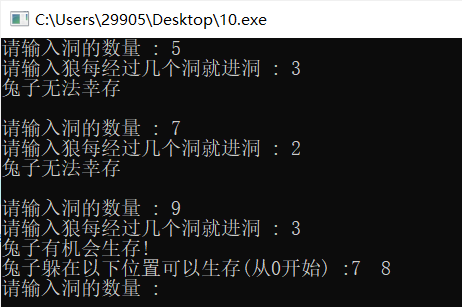
}

cout << endl;

}

}

运行截图：



第十一题

代码：

#include <iostream>

using namespace std;

int main()

{

int n, count;

while (1)

{

cout << "请输入N的数值 (1000<n<10000) :";

cin >> n;

while (n > 1)

{

n /= 5;//当前n能被5整除的数字的数量

count += n;//count增加

}

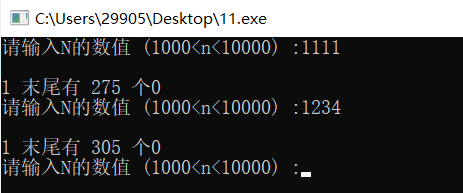
cout << endl << n << " 末尾有 " << count << " 个0" << endl;

count = 0;

}

}

运行截图：



第十二题

代码：

#include <iostream>

using namespace std;

int main()

{

int a[52], count, b{ 2 }, countUP;

for (int i = 0; i < 52; i++)

a[i] = 1;

while (count <= 104)//大循环退出条件

{

for (int i = 0; i < 52; i++)//遍历数组

{

if ((i + 1) % b == 0)//检查是否能整除

{

if (a[i] == 1)//如果为正面朝上

{

a[i] = 0;//则改为正面朝下

count++;//翻牌次数加一

if (count > 104)//如果翻牌次数大于104则退出循环

break;

}

else if (a[i] == 0)//如果为正面朝下

{

a[i] = 1;//则改为正面朝上

count++;//翻牌次数加一

if (count > 104)//如果翻牌次数大于104则退出循环

break;

}

}

}

b++;

}

cout << "第 ";

for (int i = 0; i < 52; i++)//输出值

{

if (a[i] == 1)

{

countUP++;

cout << i+1 << "、";

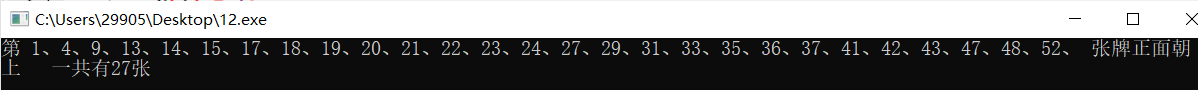
}

}

cout << " 张牌正面朝上 一共有" << countUP << "张" << endl;

}

运行截图：



第十三题

代码：

#include <iostream>

using namespace std;

int main()

{

int a, b, c, d, e;

for (a = 1; a <= 5; a++)

{

for (b = 1; b <= 5; b++)

{

if (b != a)//在上一个循环的基础上继续，但是b不能与a名次相同

{

for (c = 1; c <= 5; c++)

{

if (c != a && c != b)//在上一个循环的基础上继续，但是c不能与a,b名次相同

{

for (d = 1; d <= 5; d++)

{

if (d != a && d != b && d != c)//在上一个循环的基础上继续，但是e不能与a,b,c名次相同

{

e = 15 - a - b - c - d;//五个人的排名总数为1+2+3+4+5 可由此得出e的排名，而不需要额外的一个循环

if (((b == 3) + (c == 5) == 1) && ((d == 2) + (e == 4) == 1) && ((b == 1) + (e == 4) == 1) && ((c == 1) + (b == 2) == 1) && ((d == 2) + (a == 3) == 1))

{

cout << " a : " << a << endl << " b : " << b << endl << " c : " << c << endl << " d : " << d << endl << " e : " << e << endl;

}

}

}

}

}

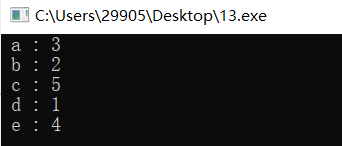
}

}

}

}

运行截图：



第十四题

代码：

#include <iostream>

using namespace std;

int main()

{

int d[3], e;

for (int i = 10; i <= 31; i++)//只有10到31之间的数字的完全平方有三位

{

e = i \* i;//临时代表完全平方数，避免修改i的值

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

{

d[j] = e % 10;//从高位开始存入

e /= 10;

}

if (d[0] == d[1] || d[0] == d[2] || d[1] == d[2])

{

for (int j = 2; j >= 0; j--)

{

cout << d[j];//从高位取出

}

cout << " ";

}

}

}

运行截图：



第十五题

代码：

#include <iostream>

using namespace std;

int main()

{

int a, b, c;

for (a = 1; a <= 3; a++)

{

for (b = 1; b <= 3; b++)

{

if (b != a)//不可能两个人跟一个人比

{

c = 6 - a - b;//总排名数减去其他两个人的排名

if (a != 1 && c != 1 && c != 3)//A不和X比，C不和X、Z比

cout << "A : " << a << " B : " << b << " C : " << c << endl;

}

}

}

}

运行截图：



第十六题

代码：

#include <iostream>

using namespace std;

int main()

{

int a, n;

while (1)

{

n = 0;//重置n的值

cout << "请输入一个数 : ";

cin >> a;

if (a % 4 == 0)

n += 1;

if (a % 7 == 0)

n += 2;

if (a % 9 == 0)

n += 4;

switch (n)//根据n的值来选择

{

case 1:

cout << "能被其中一个数 4 整除" << endl;

break;

case 2:

cout << "能被其中一个数 7 整除" << endl;

break;

case 3:

cout << "能被其中两个数 4 7 整除" << endl;

break;

case 4:

cout << "能被其中一个数 9 整除" << endl;

break;

case 5:

cout << "能被其中两个数 4 9 整除" << endl;

break;

case 6:

cout << "能被其中两个数 7 9 整除" << endl;

break;

case 7:

cout << "能同时被 4 7 9 整除" << endl;

break;

default:

cout << "不能被 4 7 9 任一个整除" << endl;

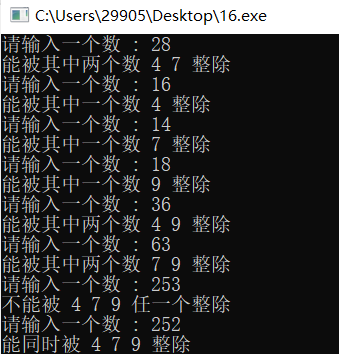
break;

}

}

}

运行截图：



第十七题

代码：

#include <iostream>

using namespace std;

int main()

{

int a, b, c, i;

while (1)

{

i = 1;

cout << "Please think of a number between 1 and 100" << endl;

cout << "Your number divided by 3 has a remainder of ";

cin >> a;

cout << "Your number divided by 5 has a remainder of ";

cin >> b;

cout << "Your number divided by 7 has a remainder of ";

cin >> c;

cout << "let me think a moment …" << endl;

for (i = 1; i < 100; i++)

{

if (i % 3 == a && i % 5 == b && i % 7 == c)//搜索条件

{

break;//找到后退出，并保留i的值

}

}

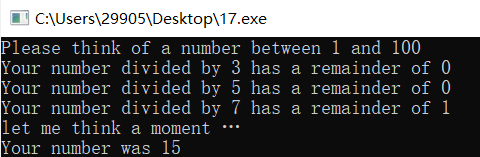
if (i != 100)//i=100时不满足以上条件

cout << "Your number was " << i << endl;

}

}

运行截图：



第十八题

代码：

#include <iostream>

using namespace std;

int main()

{

int a[10], b, c, d, f, flag = 0;;

for (b = 12345; b < 98765; b++)//五位数的最小值12345最大值98765

{

for (c = 1234; c < 9876; c++)//四位数的最小值1234最大值9876

{

if (b == c \* 2)

{

d = b;//临时替代b，c的变量，避免原值被修改

f = c;

for (int i = 0; i < 5; i++)//对五位数的每一位进行检验

{

if (a[d % 10 - 1] == 0)//如果未出现，则标记为已出现

a[d % 10 - 1] = 1;

else//如果已出现则退出循环

{

flag = 1;//标记值为1时复原数组继续外面的大循环

break;

}

d /= 10;

}

for (int i = 0; i < 4; i++)//对四位数的每一位进行检验

{

if (a[f % 10 - 1] == 0)//如果未出现，则标记为已出现

a[f % 10 - 1] = 1;

else

{

flag = 1;//标记值为1时复原数组继续外面的大循环

break;

}

f /= 10;

}

if (flag == 1)

{

for (int i = 0; i < 9; i++)//复原数组

a[i] = 0;

flag = 0;//复原标记值

continue;

}

else

cout << b << " = 2 × " << c << endl;//输出结果

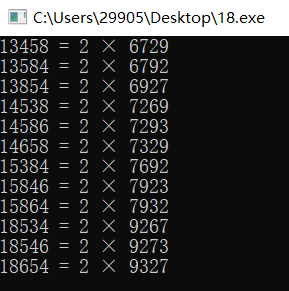
}

}

}

}

运行截图：



第十九题

代码：

#include <iostream>

#include<math.h>

#include "string.h"

using namespace std;

int main()

{

char a[50];

int b, c;

while (1)

{

c = 0;

cin >> a;

b = strlen(a);//获取数组长度

//cout << "b :" << b << endl;

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

{

//cout << "i :" << i << " Pow(16,i) :" << pow(16, i) << endl;

switch (a[i])

{

case 'a':

c += pow(16, b - (i + 1)) \* 10;//pow来源为math.h头文件，用于求出16的X次方

//cout << "c :" << c << endl;//由于读取的与实际存放的数据方向是相反的，所以需要改成b-(i+1)而不是i-1

break;

case 'b':

c += pow(16, b - (i + 1)) \* 11;

//cout << "c :" << c << endl;

break;

case 'c':

c += pow(16, b - (i + 1)) \* 12;

//cout << "c :" << c << endl;

break;

case 'd':

c += pow(16, b - (i + 1)) \* 13;

//cout << "c :" << c << endl;

break;

case 'e':

c += pow(16, b - (i + 1)) \* 14;

//cout << "c :" << c << endl;

break;

case 'f':

c += pow(16, b - (i + 1)) \* 15;

//cout << "c :" << c << endl;

break;

default:

c += pow(16, b - (i + 1)) \* (a[i] - 48);//小于10的部分

//cout << "c :" << c << endl;

break;

}

}

cout <<"转换至十进制 :"<< c << endl;

}

}

运行截图：



第二十题

代码：

#include <iostream>

using namespace std;

int main()

{

int a[10], b, c, d, tb, tc, td, flag;

for (b = 12; b <= 98; b++)//两位数循环

{

for (c = 123; c <= 987; c++)//三位数循环

{

for (d = 1234; d <= 9876; d++)//四位数循环

{

if (b \* c == d)//判断是否满足条件，如果是则继续往下，否则继续循环

{

tb = b;//避免原值被修改

tc = c;

td = d;

while (tb > 0)//取出两位数的每一位

{

if (a[(tb % 10) - 1] == 0)

a[(tb % 10) - 1] = 1;

else

flag = 1;

tb /= 10;

}

while (tc > 0)//取出三位数的每一位

{

if (a[(tc % 10) - 1] == 0)

a[(tc % 10) - 1] = 1;

else

flag = 1;

tc /= 10;

}

while (td > 0)//取出四位数的每一位

{

if (a[(td % 10) - 1] == 0)

a[(td % 10) - 1] = 1;

else

flag = 1;

td /= 10;

}

if (flag != 1)//未出现重复时输出结果

cout << b << "×" << c << "=" << d << endl;

else//否则重置数组，继续进行循环

{

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

a[i] = 0;

flag = 0;

}

}

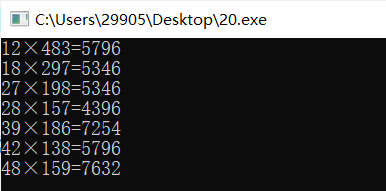
}

}

}

}

运行截图：



第二十一题

代码：

#include <iostream>

using namespace std;

int main()

{

int a[10], b, c, tb, tc, flag;

for (b = 123; b < 987; b++)

{

for (c = 123456; c < 987654; c++)

{

if (b \* b == c)

{

tb = b;//避免原值被修改

tc = c;

while (tb > 0)//取出三位数的每一位

{

if (a[(tb % 10) - 1] == 0)

a[(tb % 10) - 1] = 1;

else

flag = 1;

tb /= 10;

}

while (tc > 0)//取出六位数的每一位

{

if (a[(tc % 10) - 1] == 0)

a[(tc % 10) - 1] = 1;

else

flag = 1;

tc /= 10;

}

if (flag != 1)//未出现重复时输出结果

cout << "SQRT(" << c << ") = " << b << endl;

else//否则重置数组，继续进行循环

{

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

a[i] = 0;

flag = 0;

}

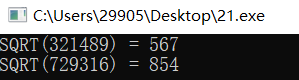
}

}

}

}

运行截图：



第二十二题

代码：

#include <iostream>

using namespace std;

int main()

{

int a[100], b, n, max;

while (1)

{

cout << "请输入N的值 : ";

cin >> n;

for (int i = 0; i < n; i++)//输入数据

cin >> a[i];

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

{

if ((a[i] + a[(i + 1) % n] + a[(i + 2) % n] + a[(i + 3) % n]) > max)//数圈判断最大的四位数

{

max = a[i] + a[(i + 1) % n] + a[(i + 2) % n] + a[(i + 3) % n];//如果比现有max大，就给max赋此四个数相加的值

b = i;//同时记下起始索引

}

}

cout << "数圈连续四位最大的数字是 : " << a[b] << " " << a[(b + 1) % n] << " " << a[(b + 2) % n] << " " << a[(b + 3) % n] << endl;

}

}

运行截图：



第二十三题

代码：

#include <iostream>

using namespace std;

int main()

{

int a, c[5], count;

while (1)

{

count = 0;

cout << "请输入一个数字(5位以内) : ";

cin >> a;

if (a == 0)//由于循环在a=0时直接跳过了，所以需要在此考虑a为0的情况

count = 1;

for (int i = 0; a > 0; i++)

{

c[i] = a % 10;//存放最低位

a /= 10;

count++;

}

cout << "此数字为 " << count << " 位数" << endl;

cout << "按序输出 : ";

for (int i = count - 1; i >= 0; i--)

{

cout << c[i] << " ";

}

cout << endl;

cout << "逆序输出 : ";

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

{

cout << c[i] << " ";

c[i] = 0;

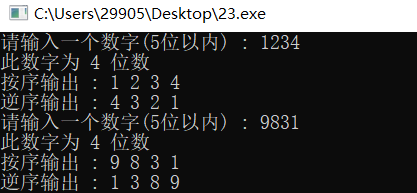
}

cout << endl;

}

}

运行截图：



第二十四题

代码：

#include <iostream>

using namespace std;

int main()

{

int a, b, c, d,e,flag;

for (a = 1; a < 10; a++)

{

for (b = 100; b < 1000; b++)

{

for (c = 1000; c < 10000; c++)

{

if (a \* b == c)

{

if (a != 2 && a != 3 && a != 5 && a != 7)//如果不满足任何条件则不为素数退出这一轮循环

continue;

d = b;

e = c;

while (d > 0)//用于确认每一位都检查到了

{

if (d % 10 != 2 && d % 10 != 3 && d % 10 != 5 && d % 10 != 7)//如果不满足任何条件则不为素数退出这一轮循环

{

flag = 1;

break;

}

d /= 10;//下一位

}

while (e > 0)//用于确认每一位都检查到了

{

if (e % 10 != 2 && e % 10 != 3 && e % 10 != 5 && e % 10 != 7)//如果不满足任何条件则不为素数退出这一轮循环

{

flag = 1;

break;

}

e /= 10;//下一位

}

if (flag == 1)//如果flag为1则复原flag的值并退出循环，不输出

{

flag = 0;

continue;

}

cout << a << " × " << b << " = " << c << endl;

}

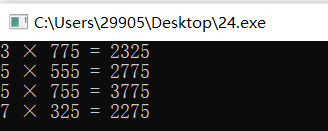
}

}

}

}

运行截图：



第二十五题

代码：

#include <iostream>

using namespace std;

int main()

{

int a, b, c, d, e, f, flag;

for (a = 100; a < 1000; a++)//被乘数

{

for (b = 10; b < 100; b++)//乘数

{

for (c = 10000; c < 100000; c++)//结果

{

if (a \* b == c)

{

d = b;

while (d > 0)//用于确认每一位都检查到了

{

if (d % 10 != 2 && d % 10 != 3 && d % 10 != 5 && d % 10 != 7)//如果不满足任何条件则不为素数退出这一轮循环

{

flag = 1;

break;

}

d /= 10;//下一位

}

e = a;

while (e > 0)//用于确认每一位都检查到了

{

if (e % 10 != 2 && e % 10 != 3 && e % 10 != 5 && e % 10 != 7)//如果不满足任何条件则不为素数退出这一轮循环

{

flag = 1;

break;

}

e /= 10;//下一位

}

f = c;

while (f > 0)//用于确认每一位都检查到了

{

if (f % 10 != 2 && f % 10 != 3 && f % 10 != 5 && f % 10 != 7)//如果不满足任何条件则不为素数退出这一轮循环

{

flag = 1;

break;

}

f /= 10;//下一位

}

d = b;

int temp1 = d % 10;

d /= 10;

int temp2 = d;

int temp3 = temp1 \* a;

int temp4 = temp2 \* a / 10;

while (temp3 > 0)

{

if (temp3 % 10 != 2 && temp3 % 10 != 3 && temp3 % 10 != 5 && temp3 % 10 != 7)//如果不满足任何条件则不为素数退出这一轮循环

{

flag = 1;

break;

}

temp3 /= 10;//下一位

}

while (temp4 > 0)

{

if (temp4 % 10 != 2 && temp4 % 10 != 3 && temp4 % 10 != 5 && temp4 % 10 != 7)//如果不满足任何条件则不为素数退出这一轮循环

{

flag = 1;

break;

}

temp4 /= 10;//下一位

}

if (flag == 1)//如果flag为1则复原flag的值并退出循环，不输出

{

flag = 0;

continue;

}

cout << a << " × " << b << " = " << c << endl;

}

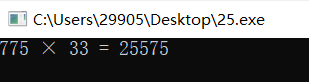
}

}

}

}

运行截图：



第二十六题

代码：

#include <iostream>

using namespace std;

int main()

{

double m, n;

int c, flag;

int a, b[15], i;

while (1)

{

cout << "请分别输入M和N的值 : ";

cin >> m >> n;

cout.precision(15);

cout << m / n;

c = m / n \* 1000000000;

while (c)

{

b[i] = c % 10;

c /= 10;

i++;

}

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

{

if (flag == 1)

break;

for (int y = i + 1; y < 15; y++)

{

if (b[i] == b[i + 1] == b[i + 2] && b[i] != 0)

{

cout << "(" << b[i] << ")" << endl;

main();

}

if (b[i] == b[y])

{

for (int d = 1; d < y - i; d++)

{

if (b[i + d] == b[y + d])

continue;

else

flag == 1;

break;

}

if (flag == 1)

break;

else

{

int d = y - i - 1;

if (b[i + d] == 0)

{

cout << endl;

main();

}

else

{

cout << "(";

for (d; d >= 0; d--)

cout << b[i + d];

cout << ")" << endl;

main();

flag == 1;

}

}

if (flag == 1)

break;

}

}

if (flag == 1)

break;

}

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

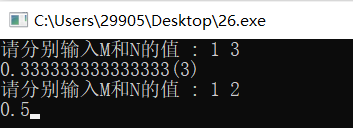
b[i] = 0;

flag = 0;

}

}

运行截图：



第二十七题

代码：

#include <iostream>

using namespace std;

int main()

{

int a;

while (1)

{

cout << " 请输入一个数字 : ";

cin >> a;

switch (a)

{

case 1:

cout << "January" << endl;

break;

case 2:

cout << "February" << endl;

break;

case 3:

cout << "March" << endl;

break;

case 4:

cout << "April" << endl;

break;

case 5:

cout << "May" << endl;

break;

case 6:

cout << "June" << endl;

break;

case 7:

cout << "July" << endl;

break;

case 8:

cout << "August" << endl;

break;

case 9:

cout << "September" << endl;

break;

case 10:

cout << "October" << endl;

break;

case 11:

cout << "November" << endl;

break;

case 12:

cout << "December" << endl;

break;

}

}

}

运行截图：

