第一题

#include <stdio.h>

void inv(int\* x, int n)

{

int\* p, temp, \* ｉ, \* ｊ;

ｉ = x;ｊ = x + n - 1;

p = x + (n - 1) / 2;

for (;ｉ <= p;ｉ++, ｊ--) /\* 调试时在该语句前设置断点 \*/

{

temp = \*ｉ;\*ｉ = \*ｊ; \*ｊ = temp;

}

/\*

1: + ｉ 0x00000055dc6ffaa8 {2} int \*

+ ｊ 0x00000055dc6ffab8 {1} int \*

+ p 0x00000055dc6ffab0 {5} int \*

2: + ｉ 0x000000ee4970fa0c {4} int \*

+ ｊ 0x000000ee4970fa14 {6} int \*

+ p 0x000000ee4970fa10 {5} int \*

.....

\*/

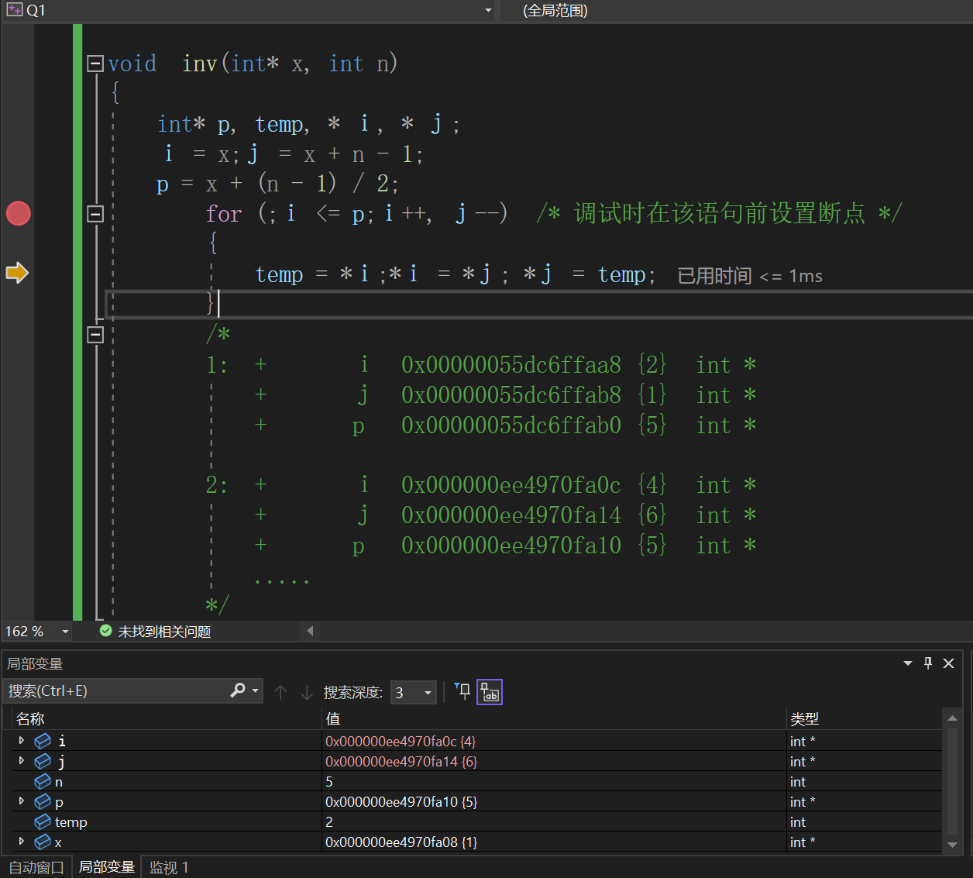
}

int main()

{

int Arr[5] = { 2,4,5,6,1 }; //初始化数组

inv(Arr, 5); //倒序

 int\* p = Arr; //利用指针输出

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

{

printf("%d", \*(p + i));

}

return 0;

}

第二题

#include <stdio.h>

void Move(char Arr[])

{

char\* Start = Arr; //定义开始/结束指针，和交换缓冲

char\* End = Arr;

char Temp;

for (;\*End != '\0'; End++); //得到结尾指针

int j = (End - Start)/2; //获得一半长度，交换上线

for (int i = 0;i < j; i++) //循环，并交换

{

Temp = \*(Start+i);

\*(Start + i) = \*(Start +j + i);

\*(Start + j + i) = Temp;

}

printf("%s", Start); //输出

}

int main()

{

char Arr[200] = "20040218brot"; //测试

Move(Arr);

return 0;

}

第三题

#include <stdio.h>

void Swap(int\* Num1, int\* Num2) //交换两个数

{

int Temp; //指针内容交换

Temp = \*Num1;

\*Num1= \*Num2;

\*Num2 = Temp;

}

void GetArr(int\* Start, int Len) //得到一个Len长的数组

{

for (int \*p = Start; p < Start + Len;p++) //利用指针循环，写入

{

scanf\_s("%d", p);

}

}

void MmExchange(int\* Start, int Len) //交换程序

{

int\* End = Start + Len - 1; //记录开始指针/结束指针，最大最小指针

int\* Max = Start;

int\* Min = Start; //假设是第一个

for (int\* p = Start; p < Start + Len; p++)

{

if (\*p > \*Max)Max = p; //如果有更大或更小则标记

if (\*p < \*Min)Min = p;

}

Swap(Start, Min); //交换

Swap(End, Max);

}

void PutArr(int\* Start, int Len) //输出

{

for (int\* p = Start; p < Start + Len;p++)

{

printf("%d ", \*p);

}

}

int main()

{

int Arr[10];

GetArr(Arr,10); //三个部分

MmExchange(Arr, 10);

PutArr(Arr, 10);

return 0;

}

第四题

#define \_CRE\_SECURE\_NO\_WARNINGS

#include <stdio.h>

void DelSpace(char\* Start)

{

char\* Read = Start; //分别定义读写指针

char\* Write = Start;

for(;\*Read != '\0';Read++)

{

if (\*Read != ' ') //全部读出，如果不是空格则写入

{

\*Write = \*Read;

Write++;

}

}

for (;\*Write != '\0';Write++)

{

\*Write = '\0'; //其余补充为\0

}

}

int main()

{

char Arr[200] = {0}; //输入字符

char\* p = Arr;

char Temp;

while(1) //循环读取直到回车键

{

Temp = getchar();

if (Temp == '\n')

{

break;

}

\*p = Temp;

p++; //待写入指针增加

}

DelSpace(Arr); //删除空格

printf("%s", Arr);

}

第五题

#include <stdio.h>

#define N 3

#define M 4

int main()

{

float Score[N][M];

float\* p = Score[0];

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

{

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

{

scanf\_s("%f", &Score[i][j]); //输入成绩

}

}

float Count = 0;

for (int j = 0; j < M; j++) //按照科目

{

float\* Max = (\*(Score + 0) + j); //假设最大最小都是第一个

float\* Min = (\*(Score + 0) + j);

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

{

Count += \*(\*(Score + i) + j); //统计总分

if (\*(\*(Score + i) + j) > \*Max) //刷新最高分最低分

{

Max = \*(Score + i) + j;

}

if (\*(\*(Score + i) + j) < \*Min)

{

Min = \*(Score + i) + j;

}

}

printf("%d 门课平均为 %f\n", j + 1, Count / (float)N);

printf("最高分 %f 最低分 %f \n", \*Max, \*Min);

Count = 0;

}

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

{

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

{

printf("%p ",(\*(Score + i) + j)); //输出地址

}

}

return 0;

}

第六题

#include <stdio.h>

int GetArr(int\* Start) //得到一个Len长的数组

{

printf("请输入数组，以0作为最后一项结束\n");

for (;;) //利用指针循环，写入

{

int Temp;

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

if (Temp) //只要不是0就写入

{

\*Start = Temp;

Start++;

}

else

{

return 0;

}

}

}

void AddArr(int\* Start1, int\* Start2, int\* Start3) //将S1，S2排序放入S3

{

for (;\*Start1 != 0 && \*Start2 != 0;Start3++) //循环到S1或S2完成

{

if (\*Start1 < \*Start2) //选择小的

{

\*Start3 = \*Start1; //写入

Start1++; //标记下一个读取的

}

else

{

\*Start3 = \*Start2; //同理

Start2++;

}

}

for (;\*Start1 != 0;Start1++,Start3++) //完成S1/S2剩下的

{

\*Start3 = \*Start1;

}

for (;\*Start2 != 0;Start2++, Start3++)

{

\*Start3 = \*Start2;

}

}

int main()

{

int Arr1[100] = { 0 }; //初始化

int Arr2[100] = { 0 };

int Arr3[100] = { 0 };

GetArr(Arr1); //输入

GetArr(Arr2);

AddArr(Arr1, Arr2, Arr3); //排序

for (int\* p = Arr3; \*p != 0; p++) //输出

{

printf("%d ", \*p);

}

}

第七题

#include <stdio.h>

int main()

{

for (int i = 0; i < 30; i++)

{

for (int j = 0; i < 30 - i; j++) //遍历可能性

{

if (3 \* i + 2 \* j + (30 - i - j) == 50) //符合要求

{

printf("男 %d，女 %d，儿童 %d",i, j, 30 - i - j); //输出

}

}

}

return 0;

}

第八题

#include <stdio.h>

#include <math.h>

int BinChar2Int(char Arr[]) //二进制char转十进制int

{

char\* Start = Arr;

char\* End = Arr;

for (;\*End != '\0';End++); //分别得到开头指针，结尾指针，和长度

int Sum = 0; //各个位和

int i = End - Start-1; //第一位的2^x

for (char\* p = Arr; p < End;p++,i--)

{

Sum += ( \* p - '0')\* pow(2, i); //计算每一位

}

return Sum;

}

int main()

{

int Block[9];

int BlockMap[8] = {0,1,2,5,8,7,6,3}; //标记矩阵读取顺序

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

{

scanf\_s("%d", &Block[i]); //输入

}

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

{

if (i == 4) //输出

{

continue;

}

if (Block[i] > Block[4])

{

Block[i] = 1;

}

else

{

Block[i] = 0;

}

}

char Result[10] = {0};

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

{

Result[i] = Block[BlockMap[i]] + '0'; //顺序读取，得到二进制

}

printf("%s", Result);

printf("\n%d", BinChar2Int(Result)); //转十进制输出

return 0;

}

第九题、

#include <stdio.h>

#include<math.h>

#define N 3

void GetHist(int psrc[], int nlen, int phist[8]) //获取N\*N矩阵

{

for (int i = 0; i < pow(nlen, 2);i++)

{

scanf\_s("%d", psrc + i); //读取

phist[\*(psrc + i)]++; //对应统计增加一次记录

}

}

void Negative(int psrc[], int pdst[], int nlen) //每一位取反

{

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

{

\*(pdst + i) = 7 - \*(psrc + i);

}

}

int main()

{

int psrc[N \* N] = {0}; //初始化

int nlen = N;

int phist[8] = { 0 };

GetHist(psrc, nlen, phist); //输入矩阵

int pdst[N \* N] = { 0 };

Negative(psrc, pdst, nlen); //取反

for (int i = 0; i < 8; i++) //输出统计信息

{

printf("%d 出现了 %d 次\n", i,\*(phist + i));

}

for (int i = 0; i < nlen \* nlen; i++) //输出取反矩阵

{

if (i % N == 0)printf("\n");

printf("%d ", \*(pdst + i));

}

return 0;

}

第十题

#include <stdio.h>

#define N 5

void Sort(int\* Scoreary, int nLen) //冒泡排序

{

int Temp;

bool Flag = true;

for (int i = 0; i <= nLen - 2 && Flag; i++)

{

Flag = false;

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

{

if (\*(Scoreary + j) > \*(Scoreary + j + 1))

{

Temp = \*(Scoreary + j);

\*(Scoreary + j) = \*(Scoreary + j + 1);

\*(Scoreary + j + 1) = Temp;

Flag = true;

}

}

}

}

void Statictic(int\* Scoreary, int nLen, int Pa[3]) //成绩分段

{

for (int i = 0; i < nLen; i++) //循环每一份成绩

{

if (\*(Scoreary + i) <= 100 && \*(Scoreary + i) >= 85) //在对应区间写入

{

Pa[0]++;

continue;

}

if (\*(Scoreary + i) <= 84 && \*(Scoreary + i) >= 60)

{

Pa[1]++;

continue;

}

if (\*(Scoreary + i) <= 59 && \*(Scoreary + i) >= 0)

{

Pa[2]++;

continue;

}

}

}

int main()

{

int ScoreArr[N]; //初始化成绩表

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

{

scanf\_s("%d", ScoreArr + i); //输入

}

Sort(ScoreArr, N); //排序

int Pa[3] = {0};

Statictic(ScoreArr, N, Pa); //分级

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

{

if (i % N == 0)printf("\n"); //输出

printf("%d ", \*(ScoreArr + i));

}

char Text[3][50] = { "100- 85分","84 - 60分" ,"59 - 0分" };

for (int i = 0; i < 3; i++) //按照分数输出

{

if (i % N == 0)printf("\n");

printf("%s \t %d \n",Text[i], \* (Pa + i));

}

}

第十一题

#include <stdio.h>

int Isparallel(float\* L1, float\* L2)

{

if (\*L1 \* \*(L2 + 1) == \*L2 \* \*(L1 + 1)) // a1\*b2 == a2\*b1条件

{

return true; //返回结构

}

return false;

}

int main()

{

float L1[3] = { 2,6,10 };

float L2[3] = { 4,12,-5 };

printf("L1 L2 %s 平行\n", Isparallel(L1, L2) ? "是":"否"); //根据结果输出

float L3[3] = { 1,2,10 };

float L4[3] = { 2,1,-5 };

printf("L3 L4 %s 平行\n", Isparallel(L3, L4) ? "是" : "否");

}