课上题目

#include <stdio.h>

void Swap1(int x, int y)

{

int tmp;

tmp = x;

x = y;

y = tmp;

}

void Swap2(int\* x, int\* y)

{

int tmp;

tmp = \*x;

\*x = \*y;

\*y = tmp;

}

void Swap3(int& x, int& y)

{

int tmp;

tmp = x;

x = y;

y = tmp;

}

int main()

{

int a1 = 2, b1 = 3;

int a2 = 2, b2 = 3;

int a3 = 2, b3 = 3;

Swap1(a1, b1);

Swap2(&a2, &b2);

Swap2(&a3, &b3);

}

第一题

#include <stdio.h>

#include <math.h>

float CalculFX(float x) //计算FX

{

return pow((x + 1), 2);

}

float CalculGX(float x) //计算GX

{

return 2 \* x + 1;

}

int main()

{

float x, y;

printf("请输入x:\n"); //获取输入

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

y = CalculFX(CalculGX(x)); //计算f（g（x））

printf("y = %f", y); //输出

return 0;

}

第二题

#include <stdio.h>

#define TestMax 10

float MaxInThree(float a, float b, float c) //获取最大值

{

float Max;

Max = a > b ? a : b;

Max = Max > c ? Max : c;

return Max;

}

int main()

{

for (float a = 0; a <= TestMax; a++)

{

for (float b = 0; b <= TestMax; b++)

{

for (float c = 0; c <= TestMax; c++) //三个循环用于测试

{

printf("%.0f %.0f %.0f 中最大 %.0f \n", a, b, c, MaxInThree(a, b, c));

}

}

}

return 0;

}

第三题

#include <stdio.h>

bool IsPrime(int Num) //判断是否是个质数

{

for (int i = 2; i <= Num / 2 + 1;i++)

{

if (Num % i == 0 && Num != i)

{

return false;

}

}

return true;

}

int main()

{

for (int i = 1; i <= 1000; i++)

{

if (IsPrime(i))

{

printf("%d 是素数\n", i);

}

}

return 0;

}

第四题

#include <stdio.h>

#include <math.h>

bool IsNarcissistic(int Num) //用于判断是否为水仙花数

{

int a, b, c, NewNum;

a = Num / 100; //获取各个位的数字

b = Num % 100 / 10;

c = Num % 10;

NewNum = pow(a, 3) + pow(b, 3) + pow(c, 3); //根据定义三次方判断

if (NewNum == Num)

{

return true;

}

return false;

}

int main()

{

for (int i = 100; i <= 999; i++)

{

if (IsNarcissistic(i))

{

printf("%d 是水仙花数\n", i);

}

}

return 0;

}

第五题

#include <stdio.h>

#include <math.h>

#define ll long long

bool IsPerfect(ll Num) //判断是否为完全数

{

ll Sum = 0; //用于记录各个因数和

for (register ll i = 1; i < Num; i++)

{

if (Num % i == 0) //循环寻找因数

{

Sum += i;

}

}

if (Sum == Num)

{

return true; //判断输出

}

return false;

}

int main()

{

for (register ll i = 1; i <= 33550336; i++)

{

if (IsPerfect(i))

{

printf("%lld 是完美数\n", i);

}

}

return 0;

}

第六题

#include <stdio.h>

bool IsLeap(int Year)

{

if ((Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0) //判断闰年条件①4整除，非100倍②400倍

{

return true;

}

return false;

}

int main()

{

for (int i = 1880; i <= 2030; i++)

{

if (IsLeap(i))

{

printf("%d 是闰年\n", i);

}

}

return 0;

}

第七题

#include <stdio.h>

bool IsLeap(int Year)

{

if ((Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0) //判断闰年条件①4整除，非100倍②400倍

{

return true;

}

return false;

}

int DayInMonth(int Year, int Month) //判断哪年那个月有几天

{

int List[] = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }; //直接打表

if (IsLeap(Year) && Month == 2)

{

return 29; //闰年二月特殊，处理29天

}

else

{

return List[Month];

}

}

int main()

{

for (int i = 1999; i <= 2030; i++)

{

for (int j = 1; j <= 12; j++)

{

printf("%5d 年的 %3d 月有 %3d 天\n", i, j, DayInMonth(i, j));

}

}

return 0;

}

第八题

#include <stdio.h>

bool IsLeap(int Year)

{

if ((Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0) //判断闰年条件①4整除，非100倍②400倍

{

return true;

}

return false;

}

int DayInMonth(int Year, int Month) //判断哪年那个月有几天

{

int List[] = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }; //直接打表

if (IsLeap(Year) && Month == 2)

{

return 29; //闰年二月特殊，处理29天

}

else

{

return List[Month];

}

}

int MonthOverDay(int Year, int Month) //判断到第几个月有多少天

{

int Count = 0;

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

{

Count += DayInMonth(Year, i); //利用之前函数计算每个月

}

return Count;

}

int main()

{

for (int i = 1999; i <= 2001; i++)

{

for (int j = 1; j <= 12; j++)

{

printf("%5d 年的 %3d 月为止有 %3d 天\n", i, j, MonthOverDay(i, j));

}

}

return 0;

}

第九题

#include <stdio.h>

bool IsLeap(int Year)

{

if ((Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0) //判断闰年条件①4整除，非100倍②400倍

{

return true;

}

return false;

}

int DayInMonth(int Year, int Month) //判断哪年那个月有几天

{

int List[] = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }; //直接打表

if (IsLeap(Year) && Month == 2)

{

return 29; //闰年二月特殊，处理29天

}

else

{

return List[Month];

}

}

int MonthOverDay(int Year, int Month) //判断到第几个月有多少天

{

int Count = 0;

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

{

Count += DayInMonth(Year, i); //利用之前函数计算每个月

}

return Count;

}

int DayOverDay(int Year, int Month, int Day) //计算这天是这年第几天

{

return MonthOverDay(Year, Month - 1) + Day; //利用前一个函数得到上个月为止少天，加上这月的天数

}

int main()

{

for (int i = 2000; i <= 2001; i++)

{

for (int j = 1; j <= 12; j++)

{

for(int k = 1; k <= 28; k++)

printf("%5d 年 %3d 月 %3d 日是该年的 %3d 天\n", i, j, k,DayOverDay(i, j, k));

}

}

return 0;

}

第十题

#include <stdio.h>

bool IsLeap(int Year)

{

if ((Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0) //判断闰年条件①4整除，非100倍②400倍

{

return true;

}

return false;

}

int DayInYear(int Year) //输出某一年有多少天

{

if (IsLeap(Year)) return 366;

return 365;

}

int DayBetweenYear(int StartYear, int EndYear) //计算年份之间的天数

{

int Count = 0;

for (int i = StartYear; i <= EndYear; i++)

{

Count += DayInYear(i); //获取每一年的，计数

}

return Count;

}

int main()

{

for (int i = 1990, j = 2010; i < j; i++, j--)

{

printf("%d 到 %d 年有 %d 天 \n", i, j, DayBetweenYear(i, j));

}

return 0;

}

第十一题

#include <stdio.h>

bool IsLeap(int Year)

{

if ((Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0) //判断闰年条件①4整除，非100倍②400倍

{

return true;

}

return false;

}

int DayInMonth(int Year, int Month) //判断哪年那个月有几天

{

int List[] = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }; //直接打表

if (IsLeap(Year) && Month == 2)

{

return 29; //闰年二月特殊，处理29天

}

else

{

return List[Month];

}

}

int MonthOverDay(int Year, int Month) //判断到第几个月有多少天

{

int Count = 0;

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

{

Count += DayInMonth(Year, i); //利用之前函数计算每个月

}

return Count;

}

int DayOverDay(int Year, int Month, int Day) //计算这天是这年第几天

{

return MonthOverDay(Year, Month - 1) + Day; //利用前一个函数得到上个月为止少天，加上这月的天数

}

int DayInYear(int Year) //输出某一年有多少天

{

if (IsLeap(Year)) return 366;

return 365;

}

int DayBetweenYear(int StartYear, int EndYear) //计算年份之间的天数

{

int Count = 0;

for (int i = StartYear; i <= EndYear; i++)

{

Count += DayInYear(i); //获取每一年的，计数

}

return Count;

}

int DayBetweenDay(int StartYear, int StartMonth, int StartDay, int EndYear, int EndMonth, int EndDay) //计算年月日-年月日的天数

{

int Count = 0;

Count = DayBetweenYear(StartYear, EndYear - 1); //先计算开始年到结束年-1的天数

Count -= DayOverDay(StartYear, StartMonth, StartDay); //减去 开始年份 到 开始日期 的天数

Count += DayOverDay(EndYear, EndMonth, EndDay); //加上 结束年 中没加上的天数

return Count;

}

int main()

{

int a = 2004, b = 2, c = 17;

int d = 2022, e = 10, f = 18;

printf("%d年%d月%d日-%d年%d月%d日，共有%d天", a, b, c, d, e, f, DayBetweenDay(a, b, c, d, e, f));

return 0;

}

第十二题

#include <stdio.h>

void Reversal(int Arr[], int len) //将数组的顺序倒换

{

int Temp;

for (int i = 0; i <= (len / 2) - 1; i++) //循环到一半，和对称位置交换

{

Temp = Arr[i];

Arr[i] = Arr[len - 1 - i];

Arr[len - 1 - i] = Temp;

}

}

int main()

{

int len;

int Arr[9999];

printf("请输入数组长度\n"); //获得数组长度

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

printf("请输入数组\n"); //输入

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

{

scanf\_s("%d", &Arr[i]);

}

Reversal(Arr, len); //顺序相反

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

{

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

}

return 0;

}

第十三题

#include <stdio.h>

void BubbleSort(int Arr[], int len) //冒泡排序

{

int temp; //交换临时空位

bool flag = true; //完成标记

for (int i = 0; i <= len - 2 && flag; i++)

{

flag = false;

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

{

if (Arr[j] < Arr[j + 1]) //比大小并交换

{

temp = Arr[j];

Arr[j] = Arr[j+1];

Arr[j + 1] = temp;

flag = true;

}

}

}

}

int main()

{

int len;

int Arr[9999] = {0};

printf("请输入数组长度\n"); //获得数组长度

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

printf("请输入数组\n");

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

{

scanf\_s("%d", &Arr[i]);

}

BubbleSort(Arr, len); //冒泡排序

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

{

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

}

return 0;

}

第十四题

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int LenOfString(char str[]) //获取字符串的长度

{

int i = 0;

for (;str[i] != '\0';i++); //带着i循环到str[i] ='/0' 为止，说明到结束符

return i;

}

int main()

{

char Str[999999];

printf("请输入一段字符串 \n");

scanf("%s", &Str);

printf("长度为%d" ,LenOfString(Str));

return 0;

}

第十五题

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int LenOfString(char str[]) //获取字符串的长度

{

int i = 0;

for (;str[i] != '\0';i++); //带着i循环到str[i] ='/0' 为止，说明到结束符

return i;

}

void AddString(char Str1[], char Str2[]) //将str2追加到str1的末尾

{

int Len1 = LenOfString(Str1); //取得长度

int Len2 = LenOfString(Str2);

for (int i = Len1, j = 0; j <= Len2; i++,j++) //从str1的结束符str1[len1]开始，用str2写入

{ //写入到str2的结束符str2[len2],不用补充结束符\0

Str1[i] = Str2[j];

}

}

int main()

{

char Str1[9999], Str2[9999];

printf("请输入两段字符串 \n");

scanf("%s", &Str1);

scanf("%s", &Str2);

AddString(Str1, Str2);

printf("%s",Str1);

return 0;

return 0;

}

第十六题

#include <stdio.h>

#define N 10

#define M 5

int main()

{

float Score[N][M] = { {99,98,97,96,95}, //成绩表

{94,93,92,91,90},

{89,88,87,86,85},

{84,83,82,81,80},

{79,78,77,76,75},

{74,73,72,71,70},

{69,68,67,66,65},

{64,63,62,61,60},

{59,58,57,56,55},

{54,53,52,51,50}, };

float sum = 0; //计数和

for (int i = 0; i <= N - 1; i++) //循环每个学生

{

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

{

sum += Score[i][j]; //科目累加

}

printf("%d 学生的平均成绩为 %.2f \n", i + 1, sum / M); //输出均值

sum = 0; //清空计数

}

printf(" \n");

for (int i = 0; i <= M - 1; i++) //循环每一门课

{

for (int j = 0; j <= N - 1; j++) //循环每个学生

{

sum += Score[j][i]; //学生累加

}

printf("%d 门课平均成绩为 %.2f \n", i + 1, sum / N); //输出

sum = 0; //清空计数

}

return 0;

}