沙河 C 语言编程题

第一次作业

1. 摄氏华氏温度转换

   【问题描述】假如用 C 表示摄氏温度，F 表示华氏温度，则有：F=C*9/5+32。输入一整数表示摄氏温度，根据该公式编程求对应的华氏温度，结果小数点后保留一位有效数字。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
	float a, b;

	scanf("%f", &b);

	a = (b * 9 / 5) + 32; // farenheit to celcius

	printf("%.1f\n", a);

	return 0;
}
```

2. 前驱、后继字符

   【问题描述】从键盘输入一个字符，求出它的前驱和后继字符（按照 ASCII 码值排序），并按照从小到大的顺序输出这三个字符和对应的 ASCII 值。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
	char a;

	scanf("%c", &a);
	printf("\n%c %c %c\n", a - 1, a, a + 1); // Display inputted character
	printf("%d %d %d\n\n", a - 1, a, a + 1); // Display ASCII Code

	return 0;

}
```

3. 人民币兑换

【问题描述】输入一个人民币的整数值（100 以内以元为单位），编程找到用 10 元、5 元、2 元、1 元表示的总数量的最小组合方式。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
	int money, tens=0, fives=0, twos=0, ones=0;

	scanf("%d", &money);

	if (money > 100) {
		printf("Input Error\n");
	}
	else
	{
		while (money >= 10)
		{
			money -= 10;
			tens += 1;
			if (money < 0)
				break;
		}
		while (money >= 5)
		{
			money -= 5;
			fives += 1;
			if (money < 0)
				break;
		}
		while (money >= 2)
		{
			money -= 2;
			twos += 1;
			if (money < 0)
				break;
		}
		while (money >= 1)
		{
			money -= 1;
			ones += 1;
			if (money < 0)
```

```
                    break;
                }
                printf("%d %d %d %d", tens, fives, twos, ones);
            }
        return 0;
}
```

4. 简易计算器

【问题描述】编程实现简易的计算器：读入两个整数运算数(data1 和 data2)及一个运算符(op)，计算表达式 data1 op data2 的值，其中 op 可以是+,-,*,/。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
    int data1, data2, res;
    float res2;
    char op;
    res = 0;

    scanf("%d %d %c", &data1, &data2, &op);

    if (op == '*')
    {
        res = data1 * data2;
        printf("%d\n", res);
    }
    else if (op == '+')
    {
        res = data1 + data2;
        printf("%d\n", res);
    }
    else if (op == '-')
    {
        res = data1 - data2;
        printf("%d\n", res);
    }
    else if (op == '/')
    {
        res2 = (float) data1 / data2;
        if (data1 % data2 == 0)
            printf("%.0f\n", res2);
    else
```

```
            printf("%.2f\n", res2);
        }
        return 0;
}
```

5. 正整数的打印

【问题描述】给出一个不多于 5 位的正整数，要求：

1.求出它是几位数。

2.分别打印出每一位数字。

3.按照逆序打印出每一位数字。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include<stdlib.h>
int main()
{
        char ent[6];
        int num, savenum = 0, rem, rev=0, dig = 0;

        fgets(ent, 6, stdin);
        num = atoi(ent);
        savenum = num + 0;
        while (num != 0)
        {
                rem = num % 10;
                rev = rev * 10 + rem;
                num /= 10;
                ++dig;
        }
        printf("%d\n%d\n%d\n", dig, savenum, rev);

        return 0;
}
```

6. 日期天数转换

【问题描述】编写一个程序，用户输入日期，计算该日期是这一年的第几天。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
	int year, month, day, count=0, pass=0, i;
	int daysperMonthl[] = { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
	int daysperMonth[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

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

	if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
	{
		for (i = 0; i < (month - 1); ++i)
			pass += daysperMonthl[i];
	}
	else
	{
		for (i = 0; i < (month - 1); ++i)
			pass += daysperMonth[i];
	}
	count += pass + day;
	printf("%d\n", count);

	return 0;
}
```

第二次作业

1. 找最大最小整数

   【问题描述】编写一个程序，用户输入若干整数，试找出其中的最大数和最小数。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <limits.h>
int main()
{
	int count, num;
	int min = INT_MAX;
	int max = INT_MIN;

	scanf("%d", &count);

	while (count--)
	{
		scanf("%d", &num);
		if (num > max)
			max = num;
		if (num < min)
			min = num;
	}
	printf("%d %d", max, min);

	return 0;
}
```

2. 反弹

【问题描述】已知一球从高空落下时，每次落地后反弹至原高度的四分之一再落下。编写一程序，从键盘输入整数 n 和 m，求该球从 n 米的高空落下后，第 m 次落地时共经过的路程以及第 m 次落地后反弹的高度，并输出结果。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
	int m, count;
	float s=0, n;

	scanf("%f %d", &n, &m);

	s += n; // original height
	for (count = 0; count < (m-1); count++)//height after divided by 4
	{
		n /= 4;
		s += n * 2;
	}
	n /= 4;
	printf("%.2f\n%.2f", s, n);

	return 0;
}
```

3. 求 A,B

【问题描述】输入三位数字 N，求两位数 AB（其中个位数字为 B，十位数字为 A，且有 0 < A < B ≤9）。使得下列等式成立：AB x BA = N。其中 BA 是把 AB 中个、十位数字交换所得的两位数。编写程序，接收控制台输入的三位整数 N，求解 A，B 并输出。如果没有解则输出“ No Answer” 。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include<stdlib.h>
int main()
{
	char ent[4];
	int num, N, a, b, t;

	fgets(ent, 4, stdin);
	num = atoi(ent);

	for (a = 1; a <= 9; a++)
	{
		for (b = 1; b <= 9; b++)
		{
			N = (a*10 + b)*((b * 10) + a);
			if (N == num)
				break;
		}
		if (N == num)
			break;
	}
	if (a>b) // A < B
	{
		t = a;
		a = b;
		b = t;
	}
	if (N == num)
		printf("%d%d", a, b);
	if (N != num)
		printf("No Answer");

	return 0;
}
```

4. 计算公式：求π 的值 b

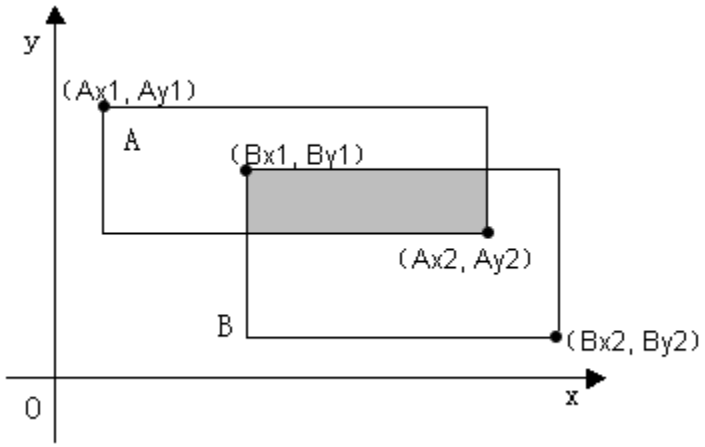
【问题描述】给定一个精度值 e，用下列公式计算π 的近似值，要求前后两次π 的迭代之差的绝对值小于 e，给出相应的最小迭代次数 n 和最后一次计算的π 的值。

π/2=1+1!/3+2!/(3×5)+3!/(3×5×7)+…+(n-1)!/(3×5×7×…×(2n-1))

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
	int n;
	double p, hp, hpt, f, e, x;

	n = 1;
	hp = 1.0;
	f = 1.0;
	x = 1.0;
	hpt = 1.0;

	scanf("%lf", &e);

	if (e < 0.000001)
		printf("Input Error");

	while (hp*2 >= e||n<2)
	{
		n++;
		f *= (n-1);
		x *= (2 * n - 1);
		hp = f / x;
		hpt += hp;
	}
	p = hpt * 2;
	printf("%d %.7lf", n, p);

	return 0;
}
```

5. 矩形相交

【问题描述】平面上有两个矩形 A 和 B，其位置是任意的。编程求出其相交部分（如图中阴影部分）的面积。（0≤a，b≤1000）




```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
int max(int m, int n)
{
	if (m>n)
		return m;
	else
		return n;
}
int min(int m, int n)
{
	if (m>n)
		return n;
	else
		return m;
}
int main()
{
	int ax1, ax2, bx1, bx2, ay1, ay2, by1, by2, aymax, aymin, bymax, bymin, axmax, axmin, bxmax, bxmin, ymin, ymax, xmin, xmax;

	scanf("%d %d %d %d", &ax1, &ay1, &ax2, &ay2);
	scanf("%d %d %d %d", &bx1, &by1, &bx2, &by2);

	aymax = max(ay1, ay2);
	aymin = min(ay1, ay2);
```

```
    axmax = max(ax1, ax2);
    axmin = min(ax1, ax2);
    bymax = max(by1, by2);
    bymin = min(by1, by2);
    bxmax = max(bx1, bx2);
    bxmin = min(bx1, bx2);
    ymin = min(aymax, bymax);
    ymax = max(aymin, bymin);
    xmin = min(axmax, bxmax);
    xmax = max(axmin, bxmin);

    if ((ymin - ymax)>0 && (xmin - xmax)>0)
        printf("%d", (ymin - ymax)*(xmin - xmax));
    else
        printf("0");

    return 0;
}
```

6. 合数分解

【问题描述】由数学基本定理可知：任何一个大于 1 的非素数整数（即合数）都可以唯一分解成若干个素数的乘积。编写程序，从控制台读入一个合数（合数的大小不会超过 int 数据类型表示的范围），求这个合数可以分解成的素数。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
    int n, i=2;

    scanf("%d", &n);

    while (n >= i)
    {
        while (n%i == 0)
        {
            printf("%d ", i);
            n /= i;
        }
        i++;
    }
    return 0;
}
```

第三次作业

1. 求差集

【问题描述】两个集合的差集定义如下：集合 A、B 的差集，由所有属于 A 但不属于 B 的元素构成。输入两个集合 A、B，每个集合中元素都是自然数。求集合 A、B 的差集。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
	int i, n, end, a, x, y;
	int A[50] = { 0 };
	int B[50] = { 0 };

	for (i = 0, end = 0; i < 50, end==0; i++)
	{
		scanf("%d", &A[i]);
		if (A[i] == -1)
			end++;
	}
	for (n = 0, end = 0; n < 50, end==0; n++)
	{
		scanf("%d", &B[n]);
		if (B[n] == -1)
			end++;
	}
	for (x = 0; x < (i-1); x++)
	{
		for (a = 0, y = 0; y < (n-1); y++)
		{
			if (A[x] != B[y])
				a++;
			if (a == (n-1))
				printf("%d ", A[x]);
		}
	}

	return 0;
}
```

2. 删数问题

【问题描述】输入一个高精度的大正整数 S（S 最长可达 240 位），去掉其中任意 N 位数字后剩下的数字按原次序组成一个新的正整数 S'。编程对给定的 N 和 S，寻找一种方案使得剩下的数字组成的新数 S' 最小。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include<string.h>
int main()
{
	char S[240];
	int N, i, s, x, y;

	scanf("%s", S);
	scanf("%d", &N);

	s = strlen(S);

	for (i = 0; i < N; i++)  //loop
	{
		for (x = 0; x < s; x++) //string checking
		{
			if (S[x] > S[x + 1]) //number changing
			{
				for (y = x; y <= s; y++)
					S[y] = S[y + 1];
				break;
			}
		}
	}
	printf("%s\n", S); //doesn't write the 0

	return 0;
}
```

3. 字母频率统计

【问题描述】编写程序从标准输入中读入一段英文，统计其中小写字母出现次数，并以柱状图的形式显示其出现次数。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
```

```
int main()
{
	char t;
	int a[26] = {0}, i, j, max;

	while (scanf("%c", &t) != EOF)
	{
		switch (t)
		{
		case'a':
			a[0]++;
			break;
		case'b':
			a[1]++;
			break;
		case'c':
			a[2]++;
			break;
		case'd':
			a[3]++;
			break;
		case'e':
			a[4]++;
			break;
		case'f':
			a[5]++;
			break;
		case'g':
			a[6]++;
			break;
		case'h':
			a[7]++;
			break;
		case'i':
			a[8]++;
			break;
		case'j':
			a[9]++;
			break;
		case'k':
			a[10]++;
			break;
		case'l':
			a[11]++;
```

```
            break;
case'm':
            a[12]++;
            break;
case'n':
            a[13]++;
            break;
case'o':
            a[14]++;
            break;
case'p':
            a[15]++;
            break;
case'q':
            a[16]++;
            break;
case'r':
            a[17]++;
            break;
case's':
            a[18]++;
            break;
case't':
            a[19]++;
            break;
case'u':
            a[20]++;
            break;
case'v':
            a[21]++;
            break;
case'w':
            a[22]++;
            break;
case'x':
            a[23]++;
            break;
case'y':
            a[24]++;
            break;
case'z':
            a[25]++;
            break;
}
```

```
        }

        max = a[0]; //only to set default

        for (i = 1; i < 26; i++) //finding the max number
        {
                if (max < a[i])
                        max = a[i];
        }

        for (i = max; i >= 1; i--)
        {
                for (j = 0; j < 26; j++)
                {
                        if (a[j] < i)
                                printf(" ");
                        else
                                printf("*");
                }

                printf("\n");
        }

        printf("abcdefghijklmnopqrstuvwxyz");

        return 0;
}
```

4. 扩展字符

【问题描述】编写一函数 expand(s1,s2)，用以将字符串 s1 中的缩记符号在字符串 s2 中扩展为等价的完整字符，例如将 a-d 扩展为 abcd。该函数可以处理大小写字母和数字，并可以处理 a-b-c、a-z0-9 与-a-z 等类似的情况。在 main 函数中测试该函数：从键盘输入包含缩记符号的字符串，然后调用该函数进行扩展，输出扩展结果。（教材 P63：Exercise 3-3）

注意：

a. 待扩展字符串中有可能包含空格，例如：a-d x-z 应扩展成：abcd xyz。所以读入待扩展字符串时，应能够读入包含空格的字符串。
b. 只要缩记符号-之后的字符比之前的字符的 ASCII 码值大，就要将它们之间的所有字符扩展出来，例如：Z-a 之间的字符也要扩展出来；
c. 特殊情况：a-b-c 将被扩展为：abc。a-a 将被扩展为：a-a。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
	int i=0, j=0;
	char t1[512], t2[512], c;

	gets(t1);

	while (t1[i] != '\0')
	{
		t2[j++] = t1[i];
		if (t1[i + 1] == '-' && t1[i+2] > t1[i])
		{
			for (c = t1[i] + 1; c <= t1[i + 2]; c++)
				t2[j++] = c;
			i += 3;
		}
		else
			i++;
	}
	t2[j] = '\0';
	puts(t2);

	return 0;
}
```

5. 矩阵运算

【问题描述】对于多个 N 阶矩阵，依次进行加、减运算。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main()
{
	int matrix[10][10];
	int n, r, c, op, num;

	scanf("%d", &n);

	for (r = 0; r < n; r++)
	{
		for (c = 0; c < n; c++)
```

```
                {
                        scanf("%d", &matrix[r][c]);
                }
        }

        while (1)
        {
                op = getchar();
                op = getchar();

                if (op == '#')
                {
                        for (r = 0; r < n; r++)
                        {
                                for (c = 0; c < n; c++)
                                {
                                        printf("%5d", matrix[r][c]);
                                }
                                printf("\n");
                        }
                        return 0;
                }

                for (r = 0; r < n; r++)
                {
                        for (c = 0; c < n; c++)
                        {
                                scanf("%d", &num);
                                if (op == '+')
                                        matrix[r][c] += num;
                                else
                                        matrix[r][c] -= num;
                        }
                }
        }
}
```

6. （文件不考）

第四次作业

1. 表达式计算（支持空格，连乘，连除）

   【问题描述】从标准输入中读入一个整数算术运算表达式，如 5 - 1 * 2 * 3 + 12 / 2 / 2 = 。计算表达式结果，并输出。要求：

   1、表达式运算符只有+、-、*、/，表达式末尾的'='字符表示表达式输入结束，表达式中可能会出现空格；

   2、表达式中不含圆括号，不会出现错误的表达式；

   3、出现除号/时，以整数相除进行运算，结果仍为整数，例如：5/3 结果应为 1。

```
ï#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
int main()
{
	int num = 0, num2, num3;
	char op = '+', op2, op3;

	while (op != '=')
	{
		scanf("%d %c", &num2, &op2);
		while (op2 == '*' || op2 == '/') //count the * or / first
		{
			scanf("%d %c", &num3, &op3);
			if (op2 == '*')
				num2 *= num3;
			else if (op2 == '/')
				num2 /= num3;
			op2 = op3;
		}
		if (op == '+')
			num += num2;
		else if (op == '-')
			num -= num2;
		op = op2;
	}
	printf("%d", num);

	return 0;
}
```

2. 超长正整数的减法

【问题描述】编写程序实现两个超长正整数（每个最长 80 位数字）的减法运算。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include<string.h>
#define MAX 81
int main()
{
	char A[MAX], B[MAX], C[MAX]={ '/0' };
	int i, j, s = 0, t, ct, m;

	gets(A);
	gets(B);

	i = strlen(A) - 1;
	j = strlen(B) - 1;

	if (i > j)
	{
		while (i >= 0)
		{
			while (j >= 0)
			{
				if (A[i] < B[j])
				{
					A[i - 1] -= 1;
					C[i] = (A[i]-'0') + 10 - (B[j]-'0') + '0';
				}
				else
					C[i] = (A[i]-'0') - (B[j]-'0') + '0';
				i--;
				j--;
			}
			C[i] = A[i] - 0;
			i--;
		}
	}
	else if (i < j)
	{
		while (j >= 0)
		{
			while (i >= 0)
			{
```

```
                        if (B[j] < A[i])
                        {
                                C[j] = (B[j]-'0') + 10 - (A[i]-'0') + '0';
                                if (B[j - 1] > '0')
                                        B[j - 1] -= 1;
                                else
                                {
                                        for (m = j - 1; B[m] == '0' && m >= 0; m--);
                                        B[m] -= 1;
                                        for (; m < j-1; m++)
                                                B[m + 1] += 9;
                                }
                        }
                        else
                                C[j] = (B[j]-'0') - (A[i]-'0') + '0';
                        i--;
                        j--;
                }

                C[j] = B[j] - 0;
                j--;
        }
        printf("-");
}

else if (i == j)
{
        for (i = j = 0; i < strlen(A); i++, j++)
        {
                if (A[i] < B[j])
                {
                        s = 1;
                        break;
                }
        }
        if (s==1)
        {
                while (j >= 0 && i >= 0)
                {
                        if (B[j] < A[i])
                        {
                                B[j - 1] -= 1;
                                C[i] = (B[j]-'0') + 10 - (A[i]-'0') + '0';
                        }
```

```
                else
                        C[i] = (B[j]-'0') - (A[i]-'0') + '0';
                i--;
                j--;
            }
            printf("-");
        }

        else if (s==0)
        {
            while (i >= 0 && j >=0)
            {
                if (A[i] < B[j])
                {
                    A[i - 1] -= 1;
                    C[i] = (A[i]-'0') + 10 - (B[j]-'0') + '0';
                }
                else
                    C[i] = (A[i]-'0') - (B[j]-'0') + '0';
                i--;
                j--;
            }
        }
    }

    for (t = 0; C[t] == '0' && t < strlen(C); t++);

    if (t == strlen(C))
        printf("0");

    else
    {
        for (ct=t; ct < strlen(C); ct++)
        {
            if (C[ct] < '1' || C[ct] > '9')
                printf("0");
            else
                printf("%c", C[ct]);
        }
    }

    return 0;
}
```

3. （文件不考）

4. 全排列数的生成

【问题描述】输入整数 N(1 <= N <= 10)，生成从 1~N 所有整数的全排列。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

void rearrange(int *arr, int a, int b)
{
	int i;
	int temp = arr[b];
	for (i = b; i > a; i--)
		arr[i] = arr[i - 1];
	arr[a] = temp;
}

void derearrange(int *arr, int a, int b)
{
	int i;
	int temp = arr[a];
	for (i = a; i < b; i++)
		arr[i] = arr[i + 1];
	arr[b] = temp;
}

void perm(int *arr, int d, int N)
{
	int i;
	if (d > N)
	{
		for (i = 0; i <= N; i++)
			printf("%d ", arr[i]);
		printf("\n");
	}
	else
	{
		for (i = d; i <= N; i++)
		{
			rearrange(arr, d, i);
			perm(arr, d + 1, N);
			derearrange(arr, d, i);
```

```
                }
        }
}

int main()
{
        int N;
        scanf("%d", &N);
        int *arr = (int *)malloc(N * sizeof(int));
        int i;
        for (i = 0; i < N; i++)
                arr[i] = i + 1;
        perm(arr, 0, N - 1);

        free(arr);
        return 0;
}
```

5. （文件不考）
6. 电话薄排序

【问题描述】编写一个程序，输入 N 个用户的姓名和电话号码，按照用户姓名的词典顺序排列输出用户的姓名和电话号码。

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#define FLUSH fflush(stdin);

typedef struct
{
        char name[11];
        char number[11];
}userinfo, *userptr;

void Sort(userinfo[]);

int n;

int main()
{
        int i, count, c=0;
```

```
	userinfo list[51] = { '\0' };

	scanf("%d", &n);

	for (count = 0; count < n; count++)
	{
		scanf("%10s", list[count].name);
		while (1)
		{
			c = getchar();
			if (c == ' ' || c == '\n' || c == EOF)
				break;
		}
		scanf("%10s", list[count].number);
		while (1)
		{
			c = getchar();
			if (c == ' ' || c == '\n' || c == EOF)
				break;
		}
	}

	Sort(list);

	for (i = 0; i < count; i++)
	{
		printf("%12s%12s", list[i].name, list[i].number);
		printf("\n");
	}

	system("pause");
	return 0;
}

void Sort(userinfo list[]) {
	userinfo temp;
	int i, j;

	for (i = 0; i<n; i++) {
		for (j = i + 1; j<n; j++) {
			if (strcmp(list[i].name, list[j].name) > 0) {
				temp = list[i];
				list[i] = list[j];
				list[j] = temp;
```

```
                        }
                    }
                }
            }
```