

第二次作业

author: Hommer

date: 0927

1. 根据3.2.4节最后的读字符代码编写完整的程序，在接收完字符后输出数组中的字符串。尝试输入尽可能长的字符串以及修改程序中的表达式，观察运行结果。

```
#include <stdio.h>

int main(){
    char s[20],c;
    int i;
    for(i=0;(i<20)&&((c=getchar())!='\n');++i)
        s[i]=c;
    printf("%s\n",s);
    return 0;
}
```

2.小明以10%的单利息投资了1000元（即，每年利息为原始本金的10%）。小红以5%的复合利息投资了1000元（即，每年的利息纳入本金）。编写一段程序，计算需要多

少年小红的资金总额才会超过小明，并显示那时两人各自的资金总额。

```
#include <stdio.h>

int main(void) {
    const double principal = 1000.0;
    const double rate_xiaoming = 0.10; // 单利（相对于原始本金）
    const double rate_xiaohong = 0.05; // 复利

    double xm = principal; // 小明资金
    double xh = principal; // 小红资金
    int years = 0;

    // 计算直到小红的资金超过小明
    while (xh <= xm) {
        years++;
        xm = principal + principal * rate_xiaoming * years;
        xh *= (1.0 + rate_xiaohong);
    }

    printf("需要 %d 年，小红的资金才会超过小明。\\n", years);
    printf("此时小明的资金：%.2f 元\\n", xm);
    printf("此时小红的资金：%.2f 元\\n", xh);

    return 0;
}
```

3.编程，求：圆周长、圆面积、圆球表面积、圆球体积、圆柱体积。要求，键盘输入圆半径及圆柱高，输出所求各结果并辅以文字说明。

```
#include <stdio.h>
# define PI 3.14

int main(void) {
    double r, h;

    printf("Please enter the circle radius: ");
    scanf("%lf", &r);

    printf("Please enter the cylinder height: ");
    scanf("%lf", &h);

    double circumference = 2.0 * PI * r;
    double circle_area = PI * r * r;
    double sphere_surface = 4.0 * PI * r * r;
    double sphere_volume = 4.0 / 3.0 * PI * r * r * r;
    double cylinder_volume = PI * r * r * h;

    printf("\nRadius: %.6f; Height: %.6f\n", r, h);
    printf("Circle circumference: %.6f\n", circumference);
    printf("Circle area: %.6f\n", circle_area);
    printf("Sphere surface area: %.6f\n", sphere_surface);
    printf("Sphere volume: %.6f\n", sphere_volume);
    printf("Cylinder volume: %.6f\n", cylinder_volume);

    return 0;
}
```

3.某位同学编写了下面的程序，请帮助他检查，尽可能多的指出其中的错误，并写出实现相应功能的正确程序。

```
include studio.h
int main() /* 该程序打印一年有多少周 */
(
int s
s:=56;
print(There are s weeks in a year.);
return 0;
```

错误

1. stdio.h标准库拼写错误
2. 引用库时没加<>或" "
3. 主函数后面没加大括号
4. 注释格式不对
5. int s后面没加 ;
6. s的赋值语句有问题
7. printf句应该为“printf("There are %d weeks in a year.", s);”
8. 没有后括号

4.将例3.3—2中的三个整型变量改成一个整形数组，再思考能否利用循环实现排序。

```
# include <stdio.h>
int main(){
    int num[3];
    int i,j,temp;
    printf("请输入三个整数(以空格分隔): ");
    scanf("%d %d %d", &num[0], &num[1], &num[2]);
    for(i=0;i<2;i++){
        for(j=0;j<2-i;j++){
            if(num[j]>num[j+1]){
                temp=num[j];
                num[j]=num[j+1];
                num[j+1]=temp;
            }
        }
    }
    printf("这三个整数按升序排列为: %d %d %d\n", num[0], num[1], num[2]);
    return 0;
}
```

5.查询我国最新个人所得税分段累进计算方案，用多重if语句编写实现个税计算器，要求保持程序循环，直到输入个

人收入值非正时结束程序。

```
#include <stdio.h>

int main() {
    double income, taxable_income, tax, net_income;

    printf("=== 个人所得税计算器（2025年最新标准） ===\n");
    printf("说明：起征点为5000元/月，输入收入≤0时退出程序。\\n");

    while (1) {
        printf("\\n请输入月收入（元）：");
        scanf("%lf", &income);

        // 检查输入是否非正，退出条件
        if (income <= 0) {
            printf("程序已退出。\\n");
            break;
        }

        // 计算应纳税所得额
        taxable_income = income - 5000;

        // 使用多重if语句计算个税
        if (taxable_income <= 0) {
            tax = 0;
        } else if (taxable_income <= 3000) {
            tax = taxable_income * 0.03 - 0;
        } else if (taxable_income <= 12000) {
            tax = taxable_income * 0.10 - 210;
        } else if (taxable_income <= 25000) {
            tax = taxable_income * 0.20 - 1410;
        } else if (taxable_income <= 35000) {
            tax = taxable_income * 0.25 - 2660;
        } else if (taxable_income <= 55000) {
            tax = taxable_income * 0.30 - 4410;
        } else if (taxable_income <= 80000) {
            tax = taxable_income * 0.35 - 7160;
        } else {
            tax = taxable_income * 0.45 - 15160;
        }
    }
}
```

```

// 确保个税不为负
if (tax < 0) {
    tax = 0;
}

// 计算税后收入
net_income = income - tax;

// 输出结果
printf("应纳税所得额: %.2f 元\n", taxable_income);
printf("应缴个人所得税: %.2f 元\n", tax);
printf("税后收入: %.2f 元\n", net_income);
}

return 0;
}

```

6.根据下列描述的条件，分别构造一个（条件）表达式：

1.number等于或大于85，但是小于100

2.number在1~9之间，但不是5

3.number不在0~10之间

1.

```
number >= 85 && number < 100
```

2.

```
number >= 1 && number <= 9 && number != 5
```

3.

```
number < 0 && number > 10
```

7.编写完整程序，实现：输入3个整数，按由小到大顺序输

出。

```
#include <stdio.h>
int main() {
    int a, b, c;
    printf("Please enter 3 integers (separated by spaces or newlines): ");
    scanf("%d %d %d", &a, &b, &c);

    int t;
    if (a > b) { t = a; a = b; b = t; }
    if (a > c) { t = a; a = c; c = t; }
    if (b > c) { t = b; b = c; c = t; }

    printf("Sorted ascending: %d %d %d\n", a, b, c);
    return 0;
}
```

8.编程，给出一个不多于5位的正整数，要求：

- 1.求出它是几位数，输出结果；
- 2.分别输出每一位数字（格式自定，例如：从高到低）；
- 3.按逆序输出各位数字，例如原数为321，应输出123。

```

#include <stdio.h>

int main() {
    int n,i;
    printf("Please enter a positive integer with up to 5 digits: ");
    scanf("%d", &n);

    if (n <= 0 || n > 99999) {
        printf("The number must be between 1 and 99999.\n");
        return 1;
    }

    int digits[5];
    int count = 0;
    int tmp = n;
    while (tmp > 0) {
        digits[count++] = tmp % 10; // store from least significant to most
        tmp /= 10;
    }

    printf("Number of digits: %d\n", count);

    // Print digits from highest to lowest
    printf("Digits (high to low): ");
    for (i = count - 1; i >= 0; --i) {
        printf("%d", digits[i]);
        if (i > 0) printf(" ");
    }
    printf("\n");

    // Print digits in reverse order (e.g., 321 -> 123)
    printf("Digits in reverse order: ");
    for (i = 0; i < count; ++i) {
        printf("%d", digits[i]);
        if (i < count - 1) printf(" ");
    }
    printf("\n");

    // Also print the reversed integer value
    int reversed = 0;
    for (i = 0; i < count; ++i) {
        reversed = reversed * 10 + digits[i];
    }
    printf("Reversed integer: %d\n", reversed);
}

```

```
    return 0;  
}
```

9.编程，实现：从键盘输入三个整数作为边长, 判断可否构成三角形？若可则计算三角形面积，否则终止。（进一步思考，是否应进行输入数据的有效性检查，例如需是正数；如何实现输入检查？再进一步地，判断何种三角形，例如

等腰，直角等，这个比较复杂，以后返回头再做。求平方根库函数为sqrt(),需#include<math.h>)

```
#include <stdio.h>
#include <math.h>

int main() {
    int a, b, c;
    printf("Please enter three positive integers as the triangle side lengths (separated by space)\n");
    scanf("%d %d %d", &a, &b, &c);

    if (a <= 0 || b <= 0 || c <= 0) {
        printf("Side lengths must be positive integers.\n");
        return 1;
    }

    /* Check triangle inequality: sum of any two sides > third side */
    if (a + b <= c || a + c <= b || b + c <= a) {
        printf("The given lengths cannot form a triangle.\n");
        return 0;
    }

    double s = (a + b + c) / 2.0;
    double area = sqrt(s * (s - a) * (s - b) * (s - c));

    printf("The lengths can form a triangle.\n");
    printf("Sides: %d, %d, %d\n", a, b, c);
    printf("Triangle area: %.6f\n", area);

    return 0;
}
```

10.编程，实现：求解一元二次方程 $ax^2+bx+c=0$ 的根。要求，用键盘输入3个系数a,b,c，考虑各种可能的情况，输出原始方程及其解（包括无解时也有提示信息输出）。（提示：对a,b,c要进行数据有效性检查；对于有无解的情况需

判定，并给出相应结果，大体上有这样几种可能：无根，1个实根，2个相等实根，2个不等实根，2个共轭复根等)

```
#include <stdio.h>
#include <math.h>

int main(void) {
    double a, b, c;
    const double EPS = 1e-12;

    printf("Enter coefficients a, b, c (separated by spaces or newlines): ");
    scanf("%lf %lf %lf", &a, &b, &c);

    printf("Equation: %g*x^2 + %g*x + %g = 0\n", a, b, c);

    if (fabs(a) < EPS) {
        // Degenerate to linear bx + c = 0
        if (fabs(b) < EPS) {
            if (fabs(c) < EPS) {
                printf("All coefficients are zero: infinite number of solutions.\n");
            } else {
                printf("No solution: 0*x = %g (inconsistent).\n", c);
            }
        } else {
            double x = -c / b;
            printf("Linear equation. Single root: x = %.12g\n", x);
        }
        return 0;
    }

    double disc = b * b - 4.0 * a * c;
    double denom = 2.0 * a;

    if (disc > EPS) {
        double sqrt_d = sqrt(disc);
        double x1 = (-b + sqrt_d) / denom;
        double x2 = (-b - sqrt_d) / denom;
        printf("Two distinct real roots:\n");
        printf("x1 = %.12g\n", x1);
        printf("x2 = %.12g\n", x2);
    } else if (fabs(disc) <= EPS) {
        double x = -b / denom;
```

```
        printf("One real (repeated) root:\n");
        printf("x = %.12g\n", x);
    } else { // disc < 0
        double sqrt_d = sqrt(-disc);
        double real = -b / denom;
        double imag = sqrt_d / fabs(denom);
        printf("Two complex conjugate roots:\n");
        printf("x1 = %.12g + %.12g i\n", real, imag);
        printf("x2 = %.12g - %.12g i\n", real, imag);
    }

    return 0;
}
```