

《计算机程序设计》实验报告

姓名： 宋亦寒 学号： PB25000200 实验日期： 2025 年 10 月 20 日

实验名称 C 程序设计入门

一、 实验目的要求

- 1. 掌握基本语句的语法；
- 2. 尝试理解复杂程序的逻辑关系与执行顺序；
- 3. 了解常用运算符的优先级；
- 2. 熟练运用 C 语言中的运算规则&控制语句

二、 实验内容

1. 实验指导书 2.2.3 自主编程练习（表达式）1，2，3，每题选做 1 小题；

```
#include<stdio.h>
int main()
{
    printf("%f", 3.5 + 1/2 + 56%10);
}
```

C:\Windows\System32\cmd.exe 9.500000
Press any key to continue . . .

```
#include<stdio.h>
int main()
{
    int b = 3, c = 4;
    printf("%d", b > c && b == c);
}
```

C:\Windows\System32\cmd.exe 0
Press any key to continue . . .

```
#include<stdio.h>
int main()
{
    int a = 2, b = 3;
    printf("%d", a += a + b);
}
```

C:\Windows\System32\cmd.exe 7
Press any key to continue . . .

2. 实验指导书 2.3.3 自主编程练习 1、2、3、4、8（3）、10，13
选做题 1：找出 1~99 之间的全部同构数。
同构数是这样一组数：它出现在平方数的右边。 例如：5 是 25 右边的数，25 是 625 右边的数，5 和 25 都是同构数。
以上题目在第三项中完成。
选做题 2：实验指导 P76 猜数字游戏

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand(time(0));
    int target = rand() % 101;
    int guess;
    printf("0到100猜数字。 \n");

    do {
        printf("\n请输入你的猜测: ");
        if (scanf("%d", &guess) != 1)
        {
            printf("输入无效, 请输入整数!\n");
            while (getchar() != '\n');
            continue;
        }

        if (guess == target)
        {
            printf("\n恭喜你猜对了! 答案是 %d\n", target);
            int rmb = rand() % 101;
            printf("恭喜获得一个%d元红包!\n", rmb);
        }
        else if (guess < target) printf("太小了, 再试试更大的数字。 \n");
        else printf("太大了, 再试试更小的数字。 \n");
    } while (guess != target);

    return 0;
}
```

0到100猜数字。

请输入你的猜测: 50
太小了, 再试试更大的数字。

请输入你的猜测: 75
太小了, 再试试更大的数字。

请输入你的猜测: 88
太小了, 再试试更大的数字。

请输入你的猜测: 94
太小了, 再试试更大的数字。

请输入你的猜测: 97
太小了, 再试试更大的数字。

请输入你的猜测: 99
恭喜你猜对了! 答案是 99
恭喜获得一个64元红包!

Press any key to continue . . . |

三、 上机程序（有注释）及运行结果

1、

```
#include <stdio.h>
int main(void)
{
    char c1, c2;
    c1 = getchar();
    c2 = getchar();

    putchar(c1);
    putchar(c2);

    // 输出两个字符的 ASCII 码
    printf("\n%d %d\n", c1, c2);

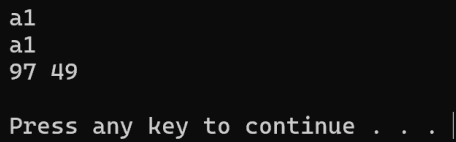
    return 0;
}
```

```
#include <stdio.h>
int main(void)
{
    char c1, c2;
    c1 = getchar();
    c2 = getchar();

    putchar(c1);
    putchar(c2);

    printf("\n%d %d\n", c1, c2);

    return 0;
}
```



```
C:\Windows\System32\cmd.exe
a1
a1
97 49
Press any key to continue . . . |
```

2、

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

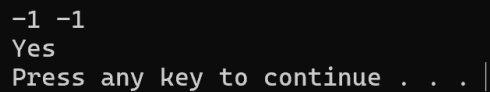
```
int main(void)
```

```
{
    int x, y;
    scanf("%d %d", &x, &y);    // 输入两个整数

    // 判断点 (x, y) 是否在方形区域  $[-2, 2] \times [-2, 2]$  内
    if (x >= -2 && x <= 2 && y >= -2 && y <= 2)
        printf("Yes");
    else
        printf("No");

    return 0;
}
```

```
#include<stdio.h>
int main(void)
{
    int x, y;
    scanf("%d %d", &x, &y);
    if (x >= -2 && x <= 2 && y >= -2 && y <= 2)
        printf("Yes");
    else
        printf("No");
    return 0;
}
```



```
C:\Windows\System32\cmd.exe
-1 -1
Yes
Press any key to continue . . . |
```

```
3、
#include<stdio.h>
int main()
{
    int n;
    for (n = 10; n < 10000; n++)
    {
        int flag1 = 0, flag2 = 1;
        int n_rev = 0, temp = n;

        // 计算 n 的反转数
        while (temp > 0)
        {
            n_rev = n_rev * 10 + temp % 10;
            temp = temp / 10;
        }

        if (n_rev == n) flag1 = 1;    // 判断是否为回文数

        // 判断是否为素数
        for (int i = 2; i * i <= n; i++)
        {
            if (n % i == 0)
            {
                flag2 = 0;
                break;
            }
        }

        // 同时满足回文和素数条件则输出
        if (flag1 && flag2)
        {
            if (n >= 10 && n < 100)
                printf("%d ", n);
            else if (n < 1000)
                printf("%d ", n);
            else
                printf("%d ", n);
        }

        // 每输出完两位或三位数范围后换行
        if (n == 99 || n == 999)
            printf("\n");
    }
}
```

```

return 0;
}

```

```

#include<stdio.h>
int main()
{
    int n;
    for (n = 10; n < 10000; n++)
    {
        int flag1 = 0, flag2 = 1;
        int n_rev = 0, temp = n;
        while (temp > 0)
        {
            n_rev = n_rev * 10 + temp % 10;
            temp = temp / 10;
        }
        if (n_rev == n) flag1 = 1;
        else flag1 = 0;

        for (int i = 2; i * i <= n; i++)
        {
            if (n % i == 0)
            {
                flag2 = 0;
                break;
            }
        }

        if (flag1 && flag2)
        {
            if (n >= 10 && n < 100)
                printf("%d ", n);
            else if (n < 1000)
                printf("%d ", n);
            else
                printf("%d ", n);
        }
    }
}

```

```

C:\Windows\System32\cmd.exe
11
101 131 151 181 191 313 353 373 383 727 757 787 797 919 929

Press any key to continue . . . |

```

4、

```

#include<stdio.h>
int main()
{
    int s1 = 0, s2 = 0, sgn = 1;
    float s3 = 0.0;
    int n, i = 1;
    scanf("%d", &n);    // 输入循环次数 n

    // for 循环计算
    for (i = 1; i <= n; i++)
    {
        s1 += i;           // 累加 1+2+...+n
        s2 += 2 * i - 1;    // 累加前 n 个奇数
        s3 += sgn * 1.0 / i; // 交错调和级数
        sgn = -sgn;         // 符号交替
    }
}

```

```
printf("for:s1=%d,s2=%d,s3=%f\n", s1, s2, s3);

// while 循环计算
s1 = 0; s2 = 0; s3 = 0.0; sgn = 1; i = 1;
while (i <= n)
{
    s1 += i;
    s2 += 2 * i - 1;
    s3 += sgn * 1.0 / i;
    sgn = -sgn;
    i++;
}
printf("while:s1=%d,s2=%d,s3=%f\n", s1, s2, s3);

// do-while 循环计算
s1 = 0; s2 = 0; s3 = 0.0; sgn = 1; i = 1;
do
{
    s1 += i;
    s2 += 2 * i - 1;
    s3 += sgn * 1.0 / i;
    sgn = -sgn;
    i++;
} while (i <= n);
printf("do while:s1=%d,s2=%d,s3=%f\n", s1, s2, s3);
}
```

```

#include<stdio.h>
int main()
{
    int s1 = 0, s2 = 0, sgn = 1;
    float s3 = 0.0;
    int n, i = 1;
    scanf("%d", &n);
    for (i = 1; i <= n; i++)
    {
        s1 += i;
        s2 += 2 * i - 1;
        s3 += sgn * 1.0 / i;
        sgn = -sgn;
    }
    printf("for:s1=%d,s2=%d,s3=%f\n", s1, s2, s3);

    s1 = 0; s2 = 0; s3 = 0.0; sgn = 1; i = 1;
    while (i <= n)
    {
        s1 += i;
        s2 += 2 * i - 1;
        s3 += sgn * 1.0 / i;
        sgn = -sgn;
        i++;
    }
    printf("while:s1=%d,s2=%d,s3=%f\n", s1, s2, s3);

    s1 = 0; s2 = 0; s3 = 0.0; sgn = 1; i = 1;
    do
    {
        s1 += i;
        s2 += 2 * i - 1;
        s3 += sgn * 1.0 / i;
        sgn = -sgn;
        i++;
    } while (i <= n);
    printf("do while:s1=%d,s2=%d,s3=%f\n", s1, s2, s3);
}

```

```

C:\Windows\System32\cmd.exe
10
for:s1=55,s2=100,s3=0.645635
while:s1=55,s2=100,s3=0.645635
do while:s1=55,s2=100,s3=0.645635
Press any key to continue . . .

```

5、

```

#include<stdio.h>
int main()
{
    int x, re;
    printf("输入一个整数:");
    scanf("%d", &x);    // 输入一个整数 x

    // 输入范围检查
    if (x < 0 || x >= 30)
    {
        printf("x 输入错误, 要求 0<=x<30\n");
        return 0;
    }

    // 根据 x 的值计算不同的 re
    switch (x / 10)
    {
        case 0: re = x; break;           // 0 <= x < 10, y = x
        case 1: re = x * x + 1; break;   // 10 <= x < 20, y = x^2 + 1
        case 2: re = x * x * x + x * x + 1; break; // 20 <= x < 30, y = x^3 + x^2 + 1
    }
    printf("x=%d 时, y=%d\n", x, re);
}

```

```

float y, ree;
printf("输入一个实数:");
scanf("%f", &y);    // 输入一个实数 y

// 实数范围检查
if (y < 0 || y >= 30)
{
    printf("x 输入错误, 要求 0<=x<30\n");
    return 0;
}
// 根据 y 的值计算不同的 ree
else if (y < 10) ree = y;           // 0 <= y < 10, y = y
else if (y < 20) ree = y * y + 1;  // 10 <= y < 20, y = y^2 + 1
else ree = y * y * y + y * y + 1;  // 20 <= y < 30, y = y^3 + y^2 + 1

printf("x=%f 时, y=%f\n", y, ree);

return 0;
}

```

The image shows a C program in a code editor and its execution in a Windows command prompt. The code defines a function that takes an integer x and a float y as input. It first checks if x is within the range [0, 30). If not, it returns 0. If x is in the range, it uses a switch statement to calculate a value 're' based on x/10. Then, it checks if y is within the range [0, 30). If not, it returns 0. If y is in the range, it uses an if-else structure to calculate 'ree' based on y. Finally, it prints the values of x, re, and y, ree.

```

#include<stdio.h>
int main()
{
    int x, re;
    printf("输入一个整数:");
    scanf("%d", &x);
    if (x < 0 || x >= 30)
    {
        printf("x输入错误, 要求0<=x<30\n");
        return 0;
    }
    switch (x / 10)
    {
        case 0: re = x; break;
        case 1: re = x * x + 1; break;
        case 2: re = x * x * x + x * x + 1; break;
    }
    printf("x=%d时, y=%d\n", x, re);

    float y, ree;
    printf("输入一个实数:");
    scanf("%f", &y);
    if (y < 0 || y >= 30)
    {
        printf("x输入错误, 要求0<=x<30\n");
        return 0;
    }
    else if (y < 10) ree = y;
    else if (y < 20) ree = y * y + 1;
    else ree = y * y * y + y * y + 1;
    printf("x=%f时, y=%f\n", y, ree);

    return 0;
}

```

Execution output in the command prompt:

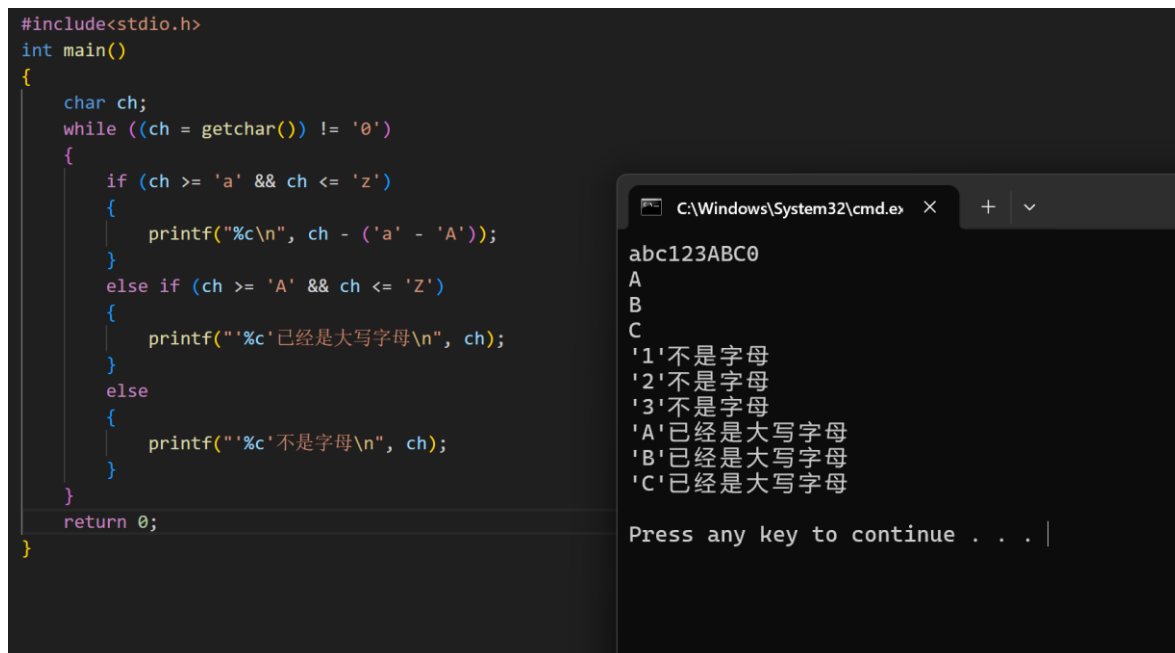
```

C:\Windows\System32\cmd.exe
输入一个整数:12
x=12时, y=145
输入一个实数:12.1
x=12.100000时, y=147.410004
Press any key to continue . . .

```


6、

```
#include<stdio.h>
int main()
{
    char ch;
    // 连续读取字符，直到输入 '0' 结束
    while ((ch = getchar()) != '0')
    {
        // 若为小写字母，转换为大写输出
        if (ch >= 'a' && ch <= 'z')
        {
            printf("%c\n", ch - ('a' - 'A'));
        }
        // 若为大写字母，提示已是大写
        else if (ch >= 'A' && ch <= 'Z')
        {
            printf("' %c' 已是大写字母\n", ch);
        }
        // 其他字符提示不是字母
        else
        {
            printf("' %c' 不是字母\n", ch);
        }
    }
    return 0;
}
```



The screenshot displays a C program in a code editor on the left and its execution output in a command prompt window on the right. The code is a program that reads characters until '0' is entered, converting lowercase letters to uppercase and providing feedback for other characters. The command prompt shows the input 'abc123ABC0' and the corresponding output: 'A', 'B', 'C', and then messages indicating that '1', '2', and '3' are not letters, while 'A', 'B', and 'C' are already uppercase letters. The prompt ends with 'Press any key to continue . . . |'.

```
#include<stdio.h>
int main()
{
    char ch;
    while ((ch = getchar()) != '0')
    {
        if (ch >= 'a' && ch <= 'z')
        {
            printf("%c\n", ch - ('a' - 'A'));
        }
        else if (ch >= 'A' && ch <= 'Z')
        {
            printf("' %c' 已是大写字母\n", ch);
        }
        else
        {
            printf("' %c' 不是字母\n", ch);
        }
    }
    return 0;
}
```

```
C:\Windows\System32\cmd.exe
abc123ABC0
A
B
C
'1'不是字母
'2'不是字母
'3'不是字母
'A'已是大写字母
'B'已是大写字母
'C'已是大写字母
Press any key to continue . . . |
```

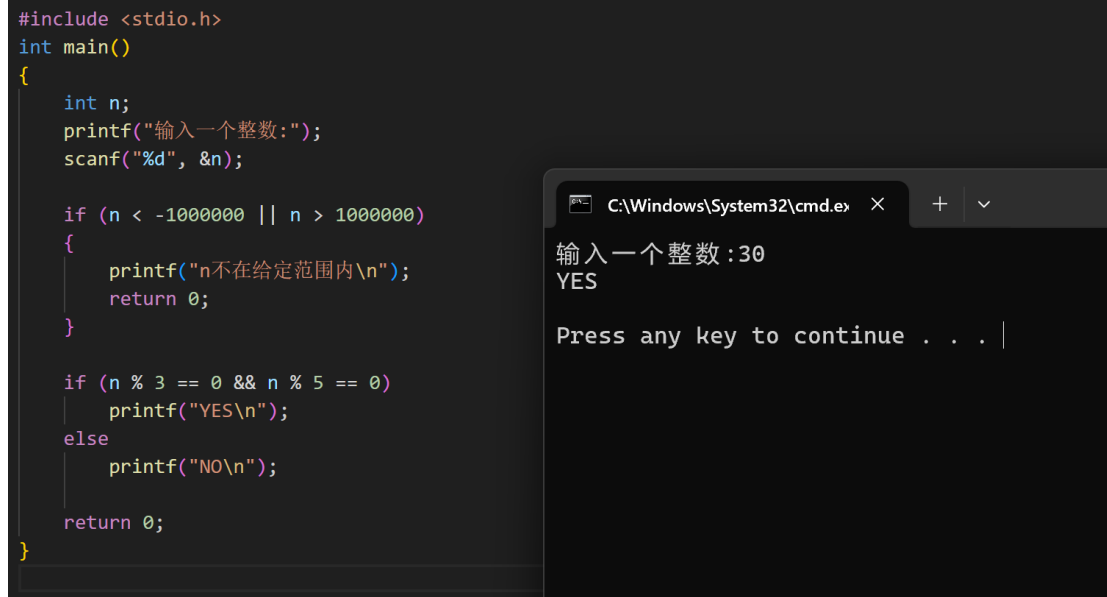
7、

```
#include <stdio.h>
int main()
{
    int n;
    printf("输入一个整数:");
    scanf("%d", &n);    // 输入整数 n

    // 判断 n 是否在有效范围内
    if (n < -1000000 || n > 1000000)
    {
        printf("n 不在给定范围内\n");
        return 0;
    }

    // 判断 n 是否同时被 3 和 5 整除
    if (n % 3 == 0 && n % 5 == 0)
        printf("YES\n");
    else
        printf("NO\n");

    return 0;
}
```



```
#include <stdio.h>
int main()
{
    int n;
    printf("输入一个整数:");
    scanf("%d", &n);

    if (n < -1000000 || n > 1000000)
    {
        printf("n不在给定范围内\n");
        return 0;
    }

    if (n % 3 == 0 && n % 5 == 0)
        printf("YES\n");
    else
        printf("NO\n");

    return 0;
}
```

C:\Windows\System32\cmd.exe × + ▾

输入一个整数:30
YES

Press any key to continue . . . |

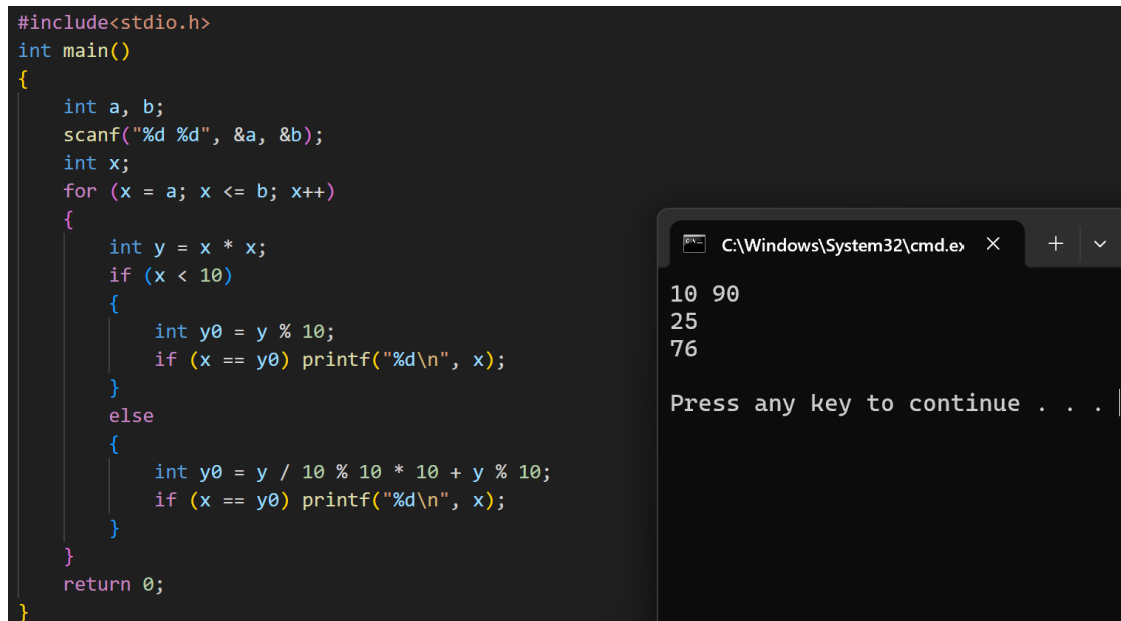
8、

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

```
scanf("%d %d", &a, &b); // 输入区间 [a, b]

int x;
for (x = a; x <= b; x++)
{
    int y = x * x; // 计算平方

    if (x < 10) // 一位数情况
    {
        int y0 = y % 10; // 取平方的个位
        if (x == y0) printf("%d\n", x);
    }
    else // 两位数情况
    {
        int y0 = y / 10 % 10 * 10 + y % 10; // 取平方的末两位
        if (x == y0) printf("%d\n", x); // 判断是否同构
    }
}
return 0;
}
```



```
#include<stdio.h>
int main()
{
    int a, b;
    scanf("%d %d", &a, &b);
    int x;
    for (x = a; x <= b; x++)
    {
        int y = x * x;
        if (x < 10)
        {
            int y0 = y % 10;
            if (x == y0) printf("%d\n", x);
        }
        else
        {
            int y0 = y / 10 % 10 * 10 + y % 10;
            if (x == y0) printf("%d\n", x);
        }
    }
    return 0;
}
```

C:\Windows\System32\cmd.exe X + v

10 90
25
76

Press any key to continue . . . |

四、 调试中的问题及解决方法（字数不限）

- 1、 在写循环计算时，我常常忘记重置变量，比如在 while 和 do while 的部分没有把 s1、s2 清零，导致结果不一致。通过不断地运行、修改、对照输出，我逐渐学会了在出现问题时耐心分析逻辑，多用 printf 调试，并且注意输入输出和变量初始化的细节。
- 2、 在判断是否为同构数时，我把取平方后末两位的表达式写错，输出结果总不对，经过一步步打印中间变量才找到原因。
- 3、 在写判断语句时，不小心把“==”写成了“=”，导致条件判断不符合预期，经过仔细检查发现了这个问题并修改。

