

第3章 迭代法

1. 用辗转相除法求两个整数的最大公约数。

C/C++代码

```
#include <stdio.h>
int main(){
    int x1, x2, a, b, c;
    scanf("%d %d", &x1, &x2);
    if (x1 < x2){
        a = x2; b = x1;
    }else{
        a = x1; b = x2;
    }
    if (b == 0){
        printf("%d 和 %d 不存在最大公约数", a, b);
    }else{
        printf("%d 和 %d 的最大公约数是", a, b);
        c = a % b;
        while(c != 0){
            a = b;
            b = c;
            c = a % b;
        }
        printf("%d", b);
    }
    return 0;
}
```

Python 代码:

```
def main():
    x1, x2 = map(int, input().split())
    if x1 < x2:
        a, b = x2, x1
    else:
        a, b = x1, x2
    if b == 0:
        print(f"{a} 和 {b} 不存在最大公约数")
    else:
```

```

    print(f"{a}和{b}的最大公约数是", end="")
    while b != 0:
        c = a % b
        a, b = b, c
    print(a)

if __name__ == "__main__":
    main()

```

2. 猴子第一天摘下若干个桃子，当即吃了一半，还不过瘾，又多吃了一个；第二天早上又将剩下的桃子吃掉一半，又多吃了一个；以此往后，到第十天早上想再吃时，就只剩一个桃子了。求第一天共摘了多少个桃子。

C/C++代码：

```

#include <stdio.h>

int main(){
    int x=1;
    for(int i=9; i>0; i--){
        x = (x+1)* 2;
    }
    printf("原有% d 个桃子", x);
    return 0;
}

```

Python 代码：

```

def main():
    x =1
    for i in range(9, 0, -1):
        x = (x +1) * 2
    print(f"原有 {x} 个桃子")

if __name__ == "__main__":
    main()

```

3. 用倒推法求杨辉三角的前 n 行并输出(直角三角形形式和等腰三角形)。

C/C++代码：

```

#include <stdio.h>

void YHSJ_RightTriangle(int n){
    int i, j, arr[100];
    i =1;
    printf("% 5d\n", i);
}

```

```

arr[1]=arr[2]=1;
printf("% 5d% 5d\n", arr[1], arr[2]);
for (i=3; i<=n; i=i+1){
    arr[1]=arr[i]=1;
    for (j=i-1; j>1; j=j-1)
        arr[j]=arr[j]+arr[j-1];
    for (j=1; j<=i; j=j+1)
        printf("% 5d", arr[j]);
    printf("\n");
}
}

void YHSJ_IsoTriangle(int n){
    int arr[100][100];
    // 初始化数组
    for (int i = 0; i < n; i++) {
        for (int j = 0; j <= i; j++) {
            if (j == 0 || j == i) {
                arr[i][j] = 1;
            }else {
                arr[i][j] = arr[i - 1][j - 1] + arr[i - 1][j];
            }
        }
    }
    // 打印杨辉三角
    for (int i = 0; i < n; i++) {
        // 打印前导空格, 使数字居中对齐
        for (int space = 0; space < n - i - 1; space++) {
            printf("    ");
        }
        // 打印数字
        for (int j = 0; j <= i; j++) {
            printf("% 5d", arr[i][j]);
            // 在数字之间打印空格(除了每行的最后一个数字)
            if (j < i) {
                printf("    ");
            }
        }
        // 打印完一行后换行
    }
}

```

```

        printf("\n");
    }
}

int main(){
    int n;
    scanf("%d", &n);
    printf("直角三角形: \n");
    YHSJ_RightTriangle(n);
    printf("等腰三角形: \n");
    YHSJ_IsoTriangle(n);
    return 0;
}

```

Python 代码:

```

def YHSJ_RightTriangle(n):
    arr = [0] * 100 # 初始化数组
    i = 1
    print(f"{i: 5d}")
    arr[0] = arr[1] = 1
    print(f"{arr[0]: 5d}{arr[1]: 5d}")

    for i in range(2, n): # 从第三行开始
        arr[0] = arr[i] = 1
        for j in range(i - 1, 0, -1):
            arr[j] = arr[j] + arr[j - 1]
        for j in range(i + 1):
            print(f"{arr[j]: 5d}", end="")
        print() # 换行

def YHSJ_IsoTriangle(n):
    arr = [[0] * 100 for _ in range(n)] # 初始化二维数组
    # 填充杨辉三角
    for i in range(n):
        for j in range(i + 1):
            if j == 0 or j == i:
                arr[i][j] = 1
            else:
                arr[i][j] = arr[i - 1][j - 1] + arr[i - 1][j]
    # 打印杨辉三角
    for i in range(n):

```

```

# 打印前导空格, 使数字居中对齐
for space in range(n - i - 1):
    print("    ", end="")
# 打印数字
for j in range(i + 1):
    print(f"{arr[i][j]: 5d}", end="")
    # 在数字之间打印空格(除了每行的最后一个数字)
    if j < i:
        print("    ", end="")
# 打印完一行后换行
print()

def main():
    n = int(input())
    print("直角三角形: ")
    YHSJ_RightTriangle(n)
    print("等腰三角形: ")
    YHSJ_IsoTriangle(n)

if __name__ == "__main__":
    main()

```