

2.4

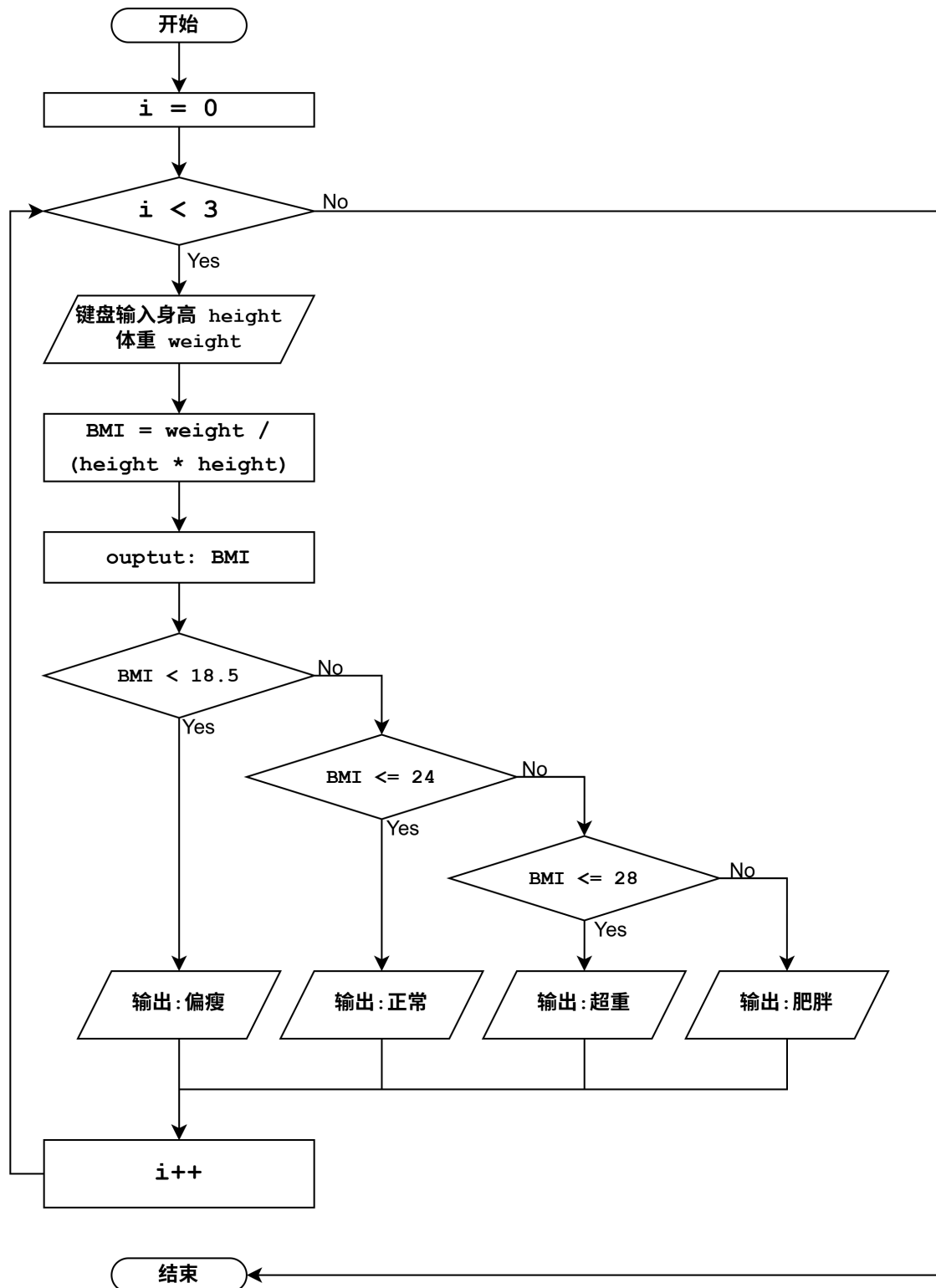
```
1  //例2.4-2 包含循环与分支语句的示例程序
2  #include <stdio.h>
3  int main(){
4      float GPA;
5      int i;
6      for(i =0;i<3;i++){
7          scanf("%f",&GPA);
8          if(GPA>4.3){
9              printf("你作弊啦!\n");
10         }
11         else{
12             printf("还要加油哦\n");
13         }
14     }
15     return 0;
16 }
```

模仿例2.4-2，编写循环多次输入身高体重值并判断身材的程序，并绘制流程图、书写伪代码。

伪代码

```
1  1. 开始
2  2. 初始化循环计数器 i = 0
3  3. 循环开始 (i 从 0 到 2)
4      a. 输入身高 (height)、体重 (weight)
5      b. 计算 BMI = weight / (height * height)
6      c. 输出 BMI 值
7      d. 如果 BMI < 18.5
8          - 输出 "身材类型: 偏瘦"
9      e. 否则如果 18.5 <= BMI < 24
10         - 输出 "身材类型: 正常"
```

11 f. 否则如果 $24 \leq \text{BMI} < 28$
12 - 输出 "身材类型: 超重"
13 g. 否则
14 - 输出 "身材类型: 肥胖"
15 4. 循环结束
16 5. 输出 "BMI计算完成! "
17 6. 结束



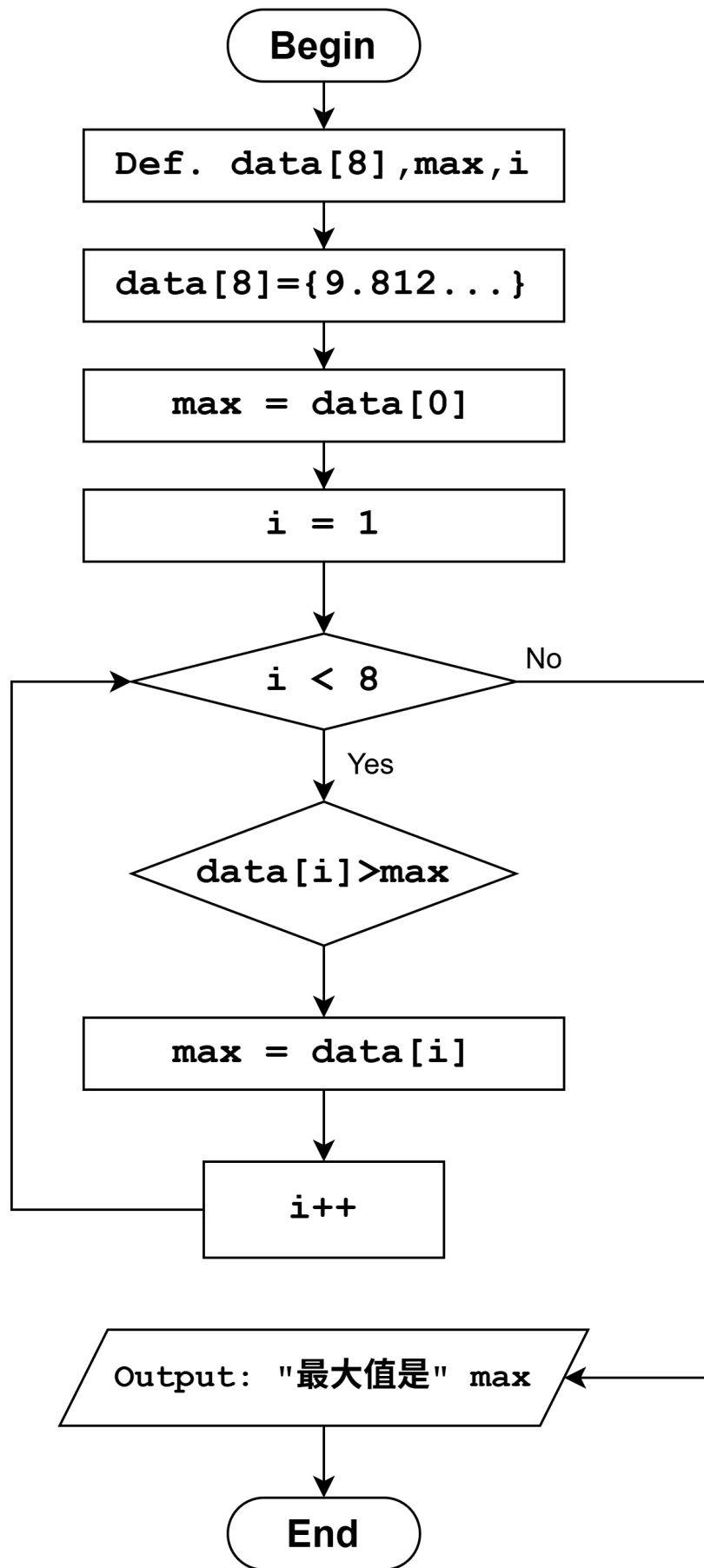
2.5

```
1 //例2.4-3 数组与循环配合求数据的平均值
2 #include <stdio.h>
3 int main(){
4     float data[8] =
5         {9.812,9.806,9.901,9.788,9.853,9.790,9.819,9.787};
6     float sum = 0;
7     float ave;
8     int i;
9     for(i = 0; i < 8; i++){
10         sum = sum + data[i];
11     }
12     ave = sum/8;
13     printf("平均值是%f", ave);
14     return 0;
15 }
```

改写例2.4-3，在循环体中使用if语句找出其中的最大值并输出，并绘制流程图、书写伪代码。

伪代码 (in English)

```
1 Find Maximum in array.
2 1. Begin,
3 2. Define array data[8] = {9.812, 9.806, 9.901, 9.788, 9.853,
4     9.790, 9.819, 9.787}
5 3. Define max, i
6 4. max = data[0]
7 5. For: i = 1 to 7
8     a. if(data[i] > max)
9         - max = data[i]
10    b. End if
11 6. Output: "最大值是" max
12 7. End.
```



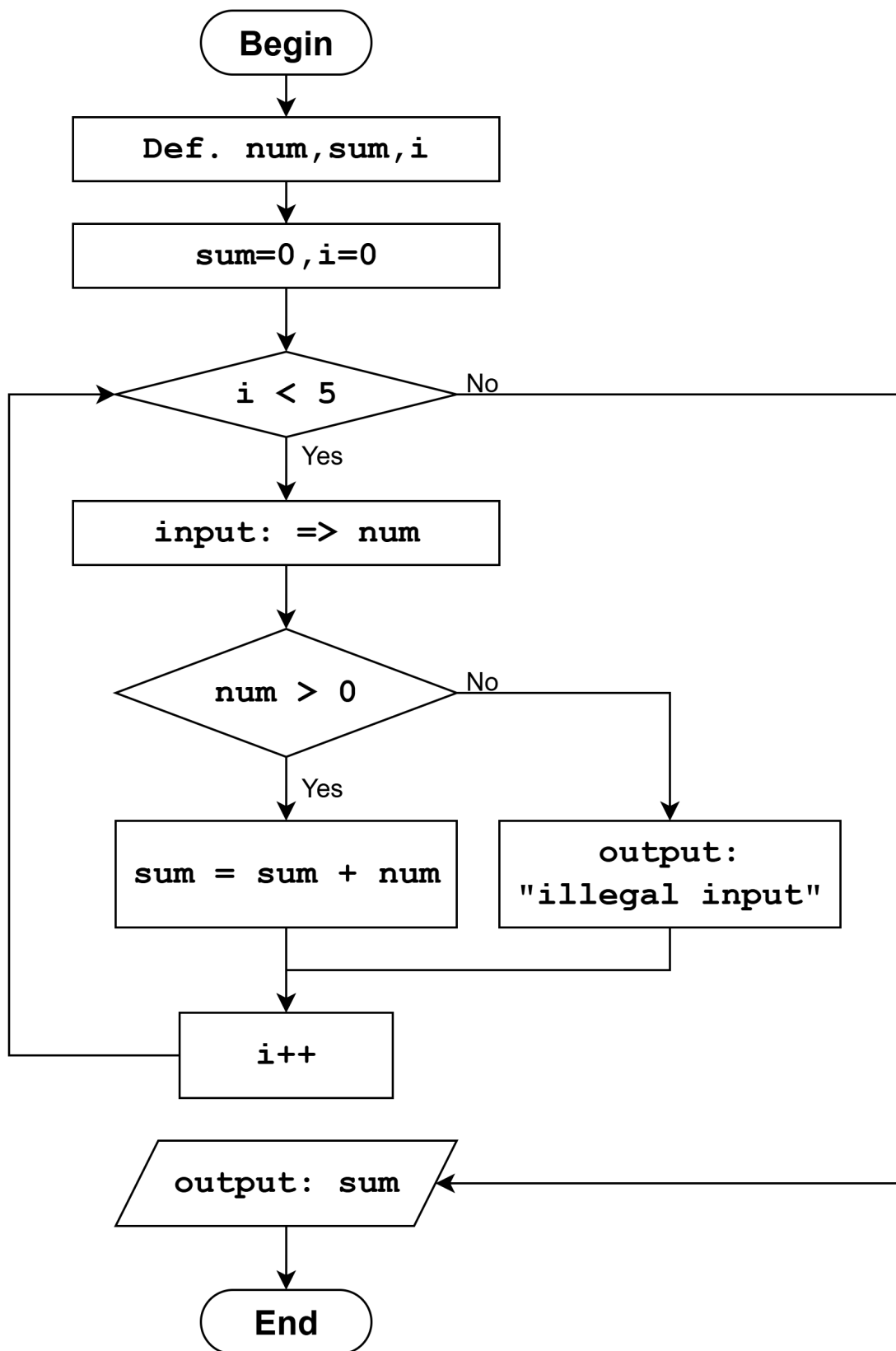
2.6

画流程图再编程，体会流程图的作用。

定义一个变量用于从键盘接收数据，一个变量用于存储累加和，使用循环语句从键盘接收5个数据，每接收一个，就判断是否为正值，并只累加正的数据，而对非正的数据提示“非法输入”。循环结束后输出累加和。运行程序后，输入的数据中包含若干非正的数据。

伪代码

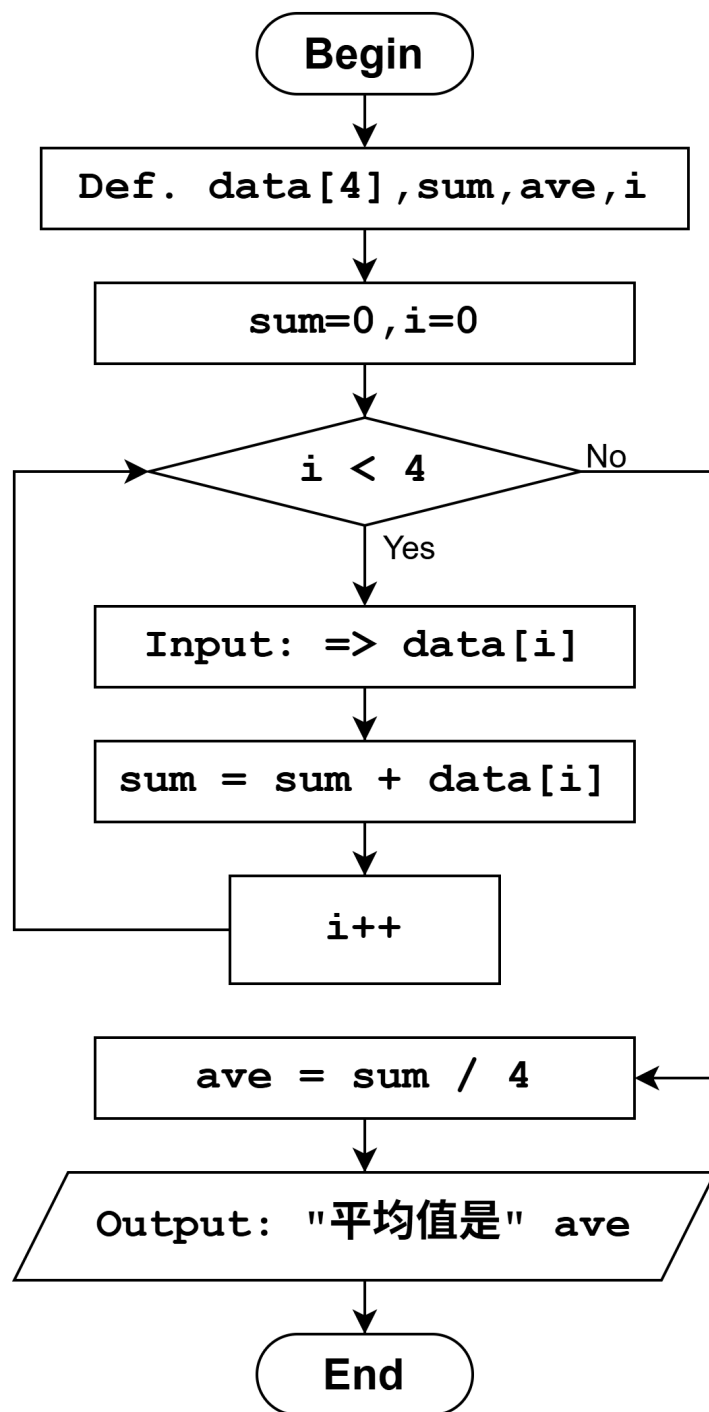
```
1  1. 开始
2  2. 定义变量 num (用于接收输入数据)
3  3. 定义变量 sum 并初始化为 0 (用于存储累加和)
4  4. 定义循环计数器 i 并初始化为 0
5  5. 循环开始: 当 i < 5 时执行循环体
6    a. 从键盘输入一个数据并赋值给 num
7    b. 如果 num > 0
8      - 将 num 加到 sum 上 (sum = sum + num)
9    c. 否则 (num <= 0)
10     - 输出 "非法输入"
11    d. 计数器 i 增加 1 (i = i + 1)
12    e. 返回循环开始判断
13 6. 循环结束
14 7. 输出累加和 sum
15 8. 结束
```



2.7

```
2  #include <stdio.h>
3  int main(){
4      float data[4];
5      float sum=0, ave;
6      int i;
7      for(i = 0; i < 4; i++){
8          scanf("%f", &data[i]);
9          sum = sum + data[i];
10     }
11     ave = sum / 4;
12     printf("平均值是%.2f", ave);
13     return 0;
14 }
```

画出例2.4-4的流程图。



2.8

先写伪代码再编程，体会伪代码的作用。

改写例2.4-4，在循环体中，`scanf` 语句后判断输入值是否为正值，并只累加正的数据。

循环结束后，计算并输出正数据的平均值。运行程序后，输入的数据中包含

若干非正的数据。

注意对程序进行额外的修改，否则运行结果不对。

修改后的代码

```
1 //例2.4-4
2 #include <stdio.h>
3 int main(){
4     float data[4];
5     float sum=0, ave;
6     int i;
7     for(i = 0; i < 4; i++){
8         scanf("%f", &data[i]);
9         if(data[i] > 0.0){
10             sum = sum + data[i];
11         }
12     }
13     ave = sum / 4;
14     printf("平均值是%.2f", ave);
15     return 0;
16 }
```

伪代码

```
1 1. 开始
2 2. 定义数组 data[4] (用于接收输入数据)
3 3. 定义变量 sum, ave_pos 并初始化为 0 (用于存储累加和, 计算平均值)
4 4. 定义循环计数器 i 并初始化为 0
5 5. 循环开始: 当 i < 4 时执行循环体
6     a. 从键盘输入一个数据并赋值给 data[i]
7     b. 如果 data[i] > 0
8         - 将 data[i] 加到 sum 上 (sum = sum + data[i])
9     c. 计数器 i 增加 1 (i = i + 1)
10    d. 返回循环开始判断
11 6. 循环结束
12 7. 计算 ave_pos = sum / 4 并输出
13 8. 结束
```

