作业十三

概念题

1. 简述命令式程序设计与声明式程序设计的区别?

```
命令式程序设计是一种通过明确的指令告诉计算机如何完成任务的编程范式。程序员需要详细描述每一步操作，控制程序的流程和状态变化。
声明式程序设计是一种描述程序逻辑而不具体描述控制流程的编程范式。程序员主要关注是什么（What）而不是如何做（How）
```

2. 函数式程序设计有哪些基本特征?

```
1. 函数是第一公民，被视为与其他变量相同的地位，可以作为参数和返回值。
2. 纯函数，并强调数据不可变性，几乎无`变量`和`状态`
3. 高阶函数，通过把函数作为参数，提升了抽象层次(map, filter, reduce)。
4. 偏函数，函数可以传入少于参数列表个数，并返回一个新的函数。
5. 函数组合，实现了流水线模型的数据流，类似链式编程。
6. 用递归实现迭代。
7. 数据被绑定到函数而不是对象，数据可以通过闭包绑定到函数中，多了一个闭包作用域。
8. 惰性求值，直到值被使用之前才会被求值。
```

3. 尾递归调用相比一般递归函数实现有什么优点?

```
减少了运行中内存的使用，避免了内存占用过大的问题，使得函数能够递归更多层而只占用一帧栈空间。
```

编程题

提交说明：请提交下面两个编程题的代码（包括测试主程序），以及运行截图

1. 猴子吃桃问题：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，且又多吃了一个。以后每天早上都吃了前一天剩下的一半且多一个。到第10天早上想再吃时，只剩下一个桃子了。

   定义下列函数

   ```
   int peach(int day);
   ```

   用于计算猴子在第 day 天的时候有几个桃子。

请你用递归、尾递归和非递归的方式实现上面的函数，并输出1-10天的桃子数。

```cpp
# include <iostream>
# include <vector>

int monkey_tail_recursion(int day, int currDay, int soFar) {
    if ( currDay == day )
        return soFar;
    return monkey_tail_recursion(day, currDay - 1, (soFar + 1) * 2);
}

int monkey_recursion(int day, int currDay) {
    if ( currDay == day )
        return 1;
    return (monkey_recursion(day, currDay - 1) + 1) * 2;
}

int monkey_iteration(int day) {
    int sum = 1;
    for ( int i = 0; i < 10 - day; i += 1 )
    {
        sum = (sum + 1) * 2;
    }
    return sum;
}

int peach(int day) {
    return monkey_tail_recursion(day, 10, 1);
}

int main(const int arg_number, const char **arg_value) {
    for ( const auto &it : std::vector<int>({1, 2, 3, 4, 5, 6, 7, 8, 9, 10}) )
    {
        std::cout << "moneky have " << peach(it) << " peaches" << "   ";
        std::cout << monkey_recursion(it, 10) << " peaches" << "   ";
        std::cout << monkey_iteration(it) << " peaches at " << "th day" << std::endl;
    }
```

```
ORMAL  us  ~/.Lectures/term2/高级程序设计/hw/hw13/monkey.cpp
moneky have 1534 peaches   1534 peaches   1534 peaches at th day
moneky have 766 peaches   766 peaches   766 peaches at th day
moneky have 382 peaches   382 peaches   382 peaches at th day
moneky have 190 peaches   190 peaches   190 peaches at th day
moneky have 94 peaches   94 peaches   94 peaches at th day
moneky have 46 peaches   46 peaches   46 peaches at th day
moneky have 22 peaches   22 peaches   22 peaches at th day
moneky have 10 peaches   10 peaches   10 peaches at th day
moneky have 4 peaches   4 peaches   4 peaches at th day
moneky have 1 peaches   1 peaches   1 peaches at th day
```

2. 请用函数式编程思想完成以下任务

现有如下数据:

```cpp
#include <string>
#include <vector>
struct S {
 std::string name;
 int population;
 std::vector<int> temperatures;
};
std::vector<S> data = {
  {
    .name = "Nanjing",
    .population = 120985,
    .temperatures = {-3, 0, 28, 29, 35, 34, 33, 32}
  },
  {
    .name = "Shanghai",
    .population = 283960,
    .temperatures = {14, 15, 24, 34, 27, 26, 10, 11, 12}
  },
```

```
    {
        .name = "Hangzhou",
        .population = 65536,
        .temperatures = {15, 17, 18, 25, 3, 31}
    }
};
```

请利用函数式编程思想实现一个算法：将人口数量和平均温度之间的关系抽取出来，返回一个二维数组表示的
结果，其值如下。每一行 (result[i]) 表示一对关系。

```
vector<vector<double>> result = {
    {120985, 23.5},
    {283960, 19.2222},
    {65536, 18.1667}
}
```

具体而言，你需要实现如下函数：

```
vector<vector<double>> extract(const vector<S> & data);
```

提示：

- 你可以使用任意STL库函数，调用任意对象提供的任意接口和任何基本运算；但你程序的控制结构只能使用分支(if, else)和递归。
- 事实上，你仅需要知道编写递归函数便可完成该任务
- 请确保你实现算法中出现函数都满足函数式程序设计的基本要求（特征）
- 你可以思考为什么函数 extract 的参数类型为常引用

提交注意事项

截止时间：2024-6-4 23:59

文件格式：姓名-学号.pdf

提交方式：南大计科在线实验教学平台

请同学们于截止时间前在南大计科在线实验教学平台上提交，每次作业最终只需要提交一个pdf文件即可，以“姓名-学号.pdf“的方式命名。代码请附在pdf中。

注意：请按要求命名文件，并且只提交一个PDF文件。任何错误的命名和文件格式将影响你的作业得分。