高级程序设计与实验课后练习十一

1. 给定一个包含 n 个**不重复**正整数的数组（例如 vector<int> nums = {1, 2, 3, 4, 5}），请找出所有大小为 k 的组合。**附加限制：** 你挑选的元素在原数组中**不能是相邻的**。

例如，若数组为 {1, 2, 3, 4, 5}，找大小为 2 的组合：

- {1, 3} 是合法的。
- {1, 2} 是**不合法**的（因为 1 和 2 在原数组中相邻）。

请编写一个递归函数 void findNonAdjacentCombinations(const vector<int> & A) 来输出所有合法的组合。

2. 给定一个只包含**正整数**的无序数组 nums（例如 {2, 3, 6, 7}），以及一个目标整数 target（例如 8）。请找出数组中所有数字的组合，使得这些数字的**和**正好等于 target。 **注意：**

1. 数组中的同一个数字在每个组合中只能使用一次。
2. 你的算法不仅要正确找到所有组合，还应尽量**高效。**在递归搜索时想办法提前中止掉那些 “显然不可能和为 target”的分支。

3. 给定一个包含 n 个**正整数**的无序数组 nums（例如 {2, 3, 6, 7}），输出可以表达为这个数组中的某子集之和的所有整数。在上面这个例子中，输出应为 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 18. 若数组中最大的整数为 W，你的函数的运行时间应为 $O(n^2W)$.

4. 你正在开发一个聊天软件的敏感词过滤系统。给定一个包含一段对话的 vector<string>，你需要将其中的特定“脏话”剔除。

但为了让净化效果更彻底，如果两个相同的脏话**紧挨着出现**，不仅要把这两个脏话删掉，还要在它们原本的位置**插入一个警告标语** "***[WARNING]***"。

任务要求： 请编写一个函数 cleanChatLog，完成上述逻辑。 **强制要求：**

1. 只能遍历该 vector **一次**。
2. 必须且只能通过**迭代器**（vector<string>::iterator）来完成遍历、判断、删除（erase）和插入（insert）操作。严禁使用下标（如 chatLog[i]）！

void cleanChatLog(vector<string>& chatLog, const string& badWord)

- **安全驾驶指南 1：** 当你使用 it = chatLog.erase(it) 删除一个元素时，it 会自动指向被删除元素的**下一个元素**。如果在 for 循环里又执行了一次 it++，你会错过什么？

- **安全驾驶指南 2：** chatLog.insert(it, "***[WARNING]***") 会返回一个迭代器，它指向的是刚刚插入的这个新元素。插入之后，为了继续往下检查，你的迭代器应该跳到哪里？

- **超前探路：** 要判断当前脏话和下一个词是不是相同的脏话，你可能需要查看 it 的下一个位置。你可以使用 next(it) 来安全地获取下一个位置的迭代器，但请务必先确认 it 还没走到 chatLog.end() - 1！