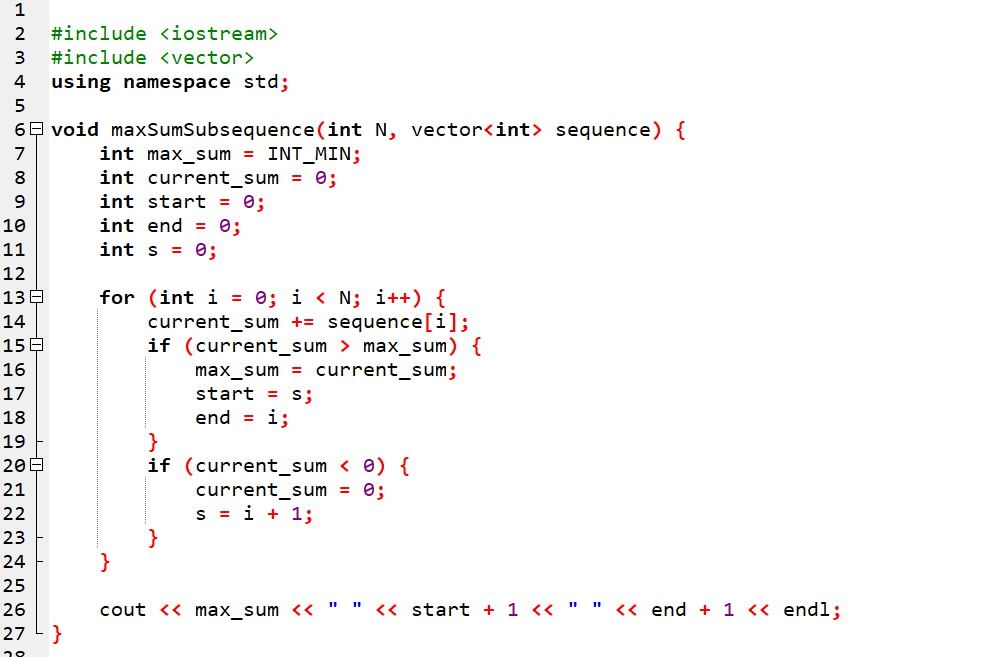
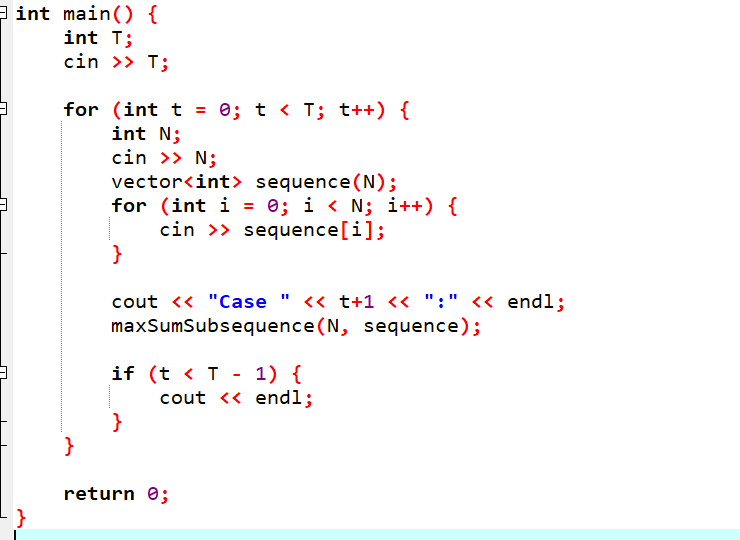
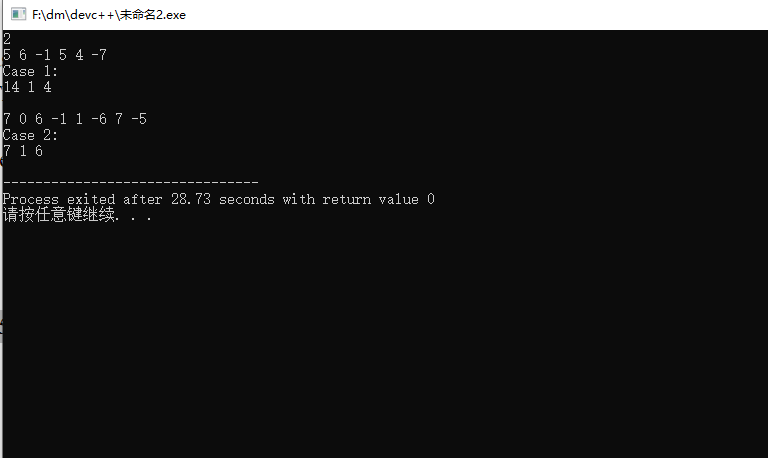
第一题

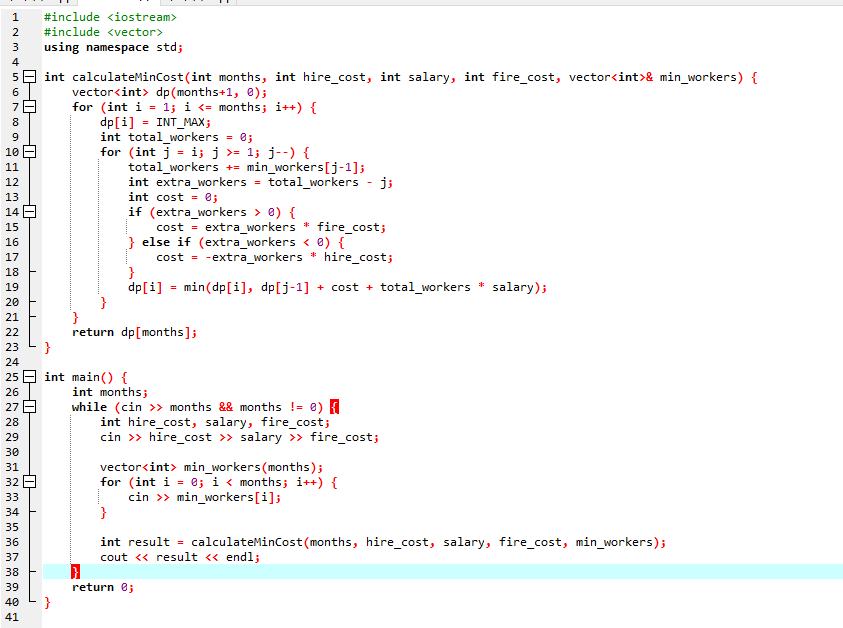


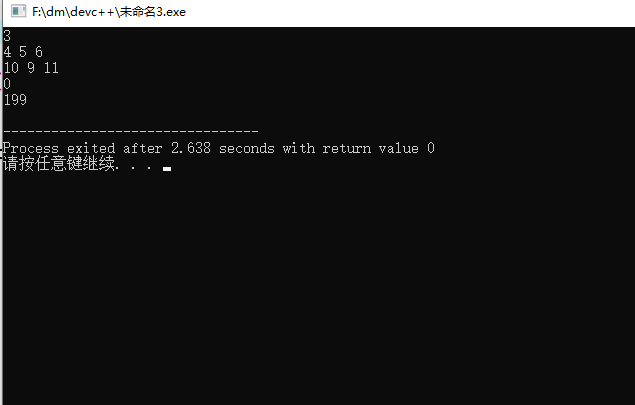




基本思路是遍历整个序列，同时维护两个变量：当前的最大子序列和 current\_sum 和全局的最大子序列和 max\_sum。在遍历过程中，对于每个元素，我们都将其加入当前的子序列和中，并判断是否超过了之前的最大和，如果是则更新 max\_sum，同时更新子序列的起始位置 start 和结束位置 end。当当前子序列和为负数时，我们将其重置为0，并更新起始位置 s。通过这种方式，我们能够在一次遍历中找到最大的子序列和及其起始和结束位置。

3：





初始化大小为 months+1 的动态规划数组 dp，用于存储每个月的最小成本。

循环从第一个月到总月份数。

对于每个月 i，将 dp[i] 初始化为无穷大 (INT\_MAX)，并跟踪所需工人的总数。

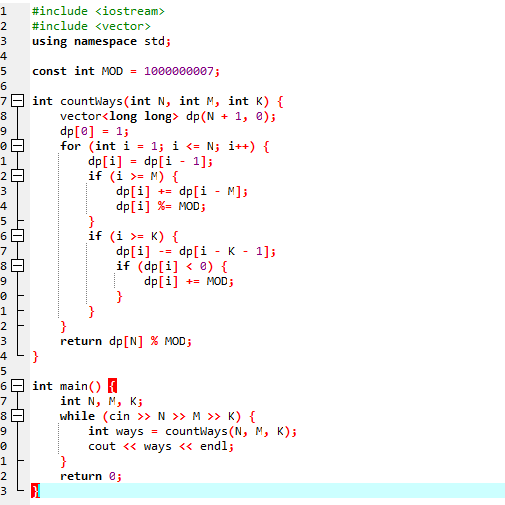
然后循环遍历之前的月份（从 j = i 到 j = 1）。

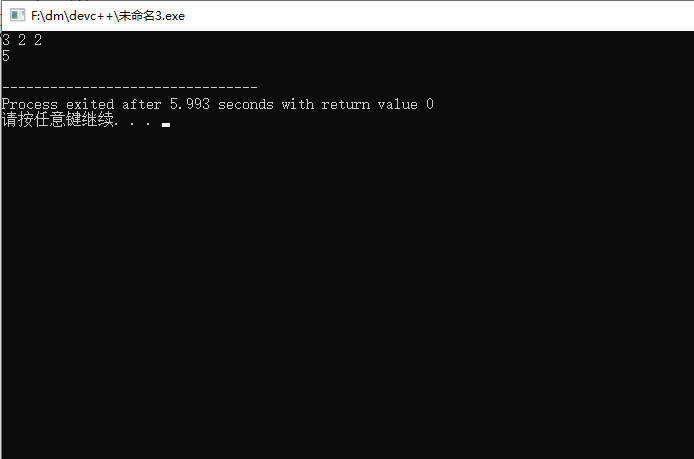
对于每个月，计算额外工人的总数（extra\_workers），并计算雇佣或解雇工人所需的成本。

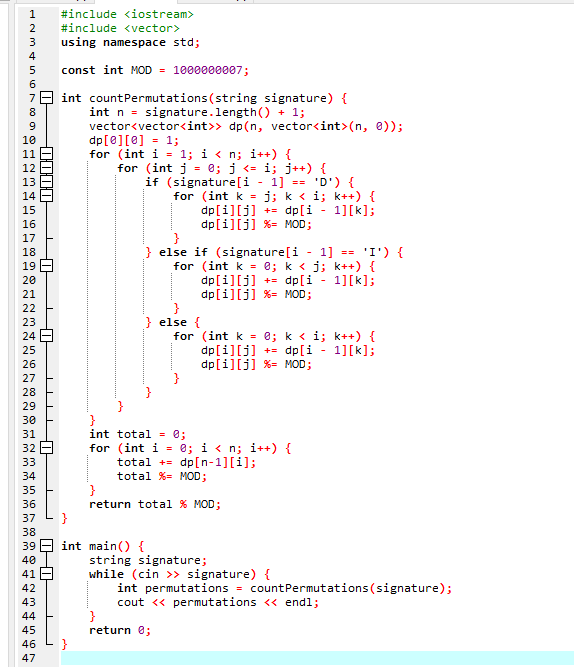
更新 dp 数组中当前月的最小成本。

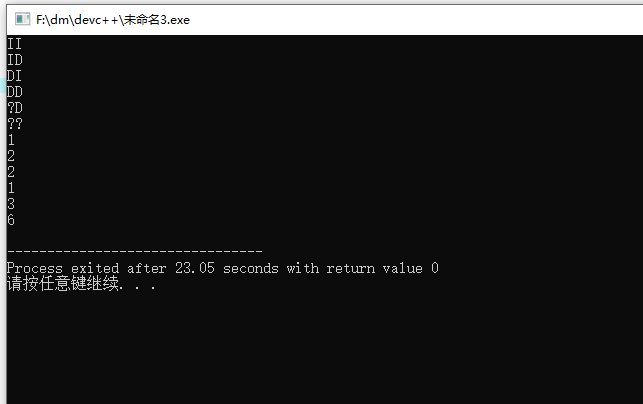
最终结果存储在 dp[months] 中，表示整个项目的最小成本。

4：

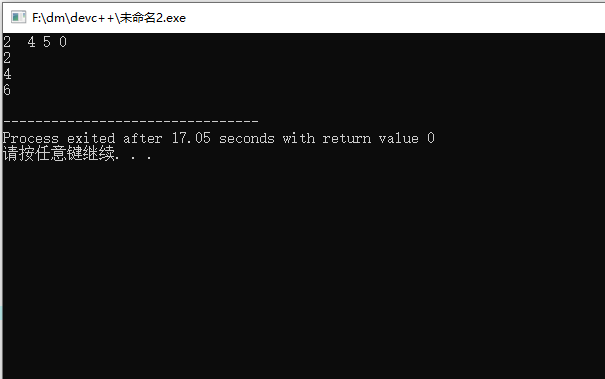


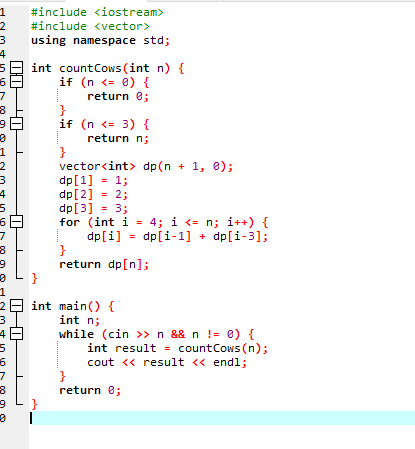


6：



9：





算法思路：

我们可以定义一个数组 dp，其中 dp[i] 表示第 i 年时母牛的数量。根据题目描述，前三年的母牛数量分别是1、2、3，然后从第四年开始，每年的母牛数量可以由前一年和前三年的母牛数量之和得到（因为第四年的母牛开始生小母牛）。因此，我们可以使用递推关系 dp[i] = dp[i-1] + dp[i-3] 来计算出第 i 年的母牛数量。