

## 编程题 1：完全平方数（动态规划）

### 算法描述：

1. 创建数组  $dp[0..n]$ ,  $dp[i]$  表示和为  $i$  的最少平方数个数
2. 初始化  $dp[0] = 0$
3. 对于每个  $i$  从 1 到  $n$ :
  - 遍历所有平方数  $j*j \leq i$
  - $dp[i] = \min(dp[i], dp[i - j*j] + 1)$
4. 返回  $dp[n]$

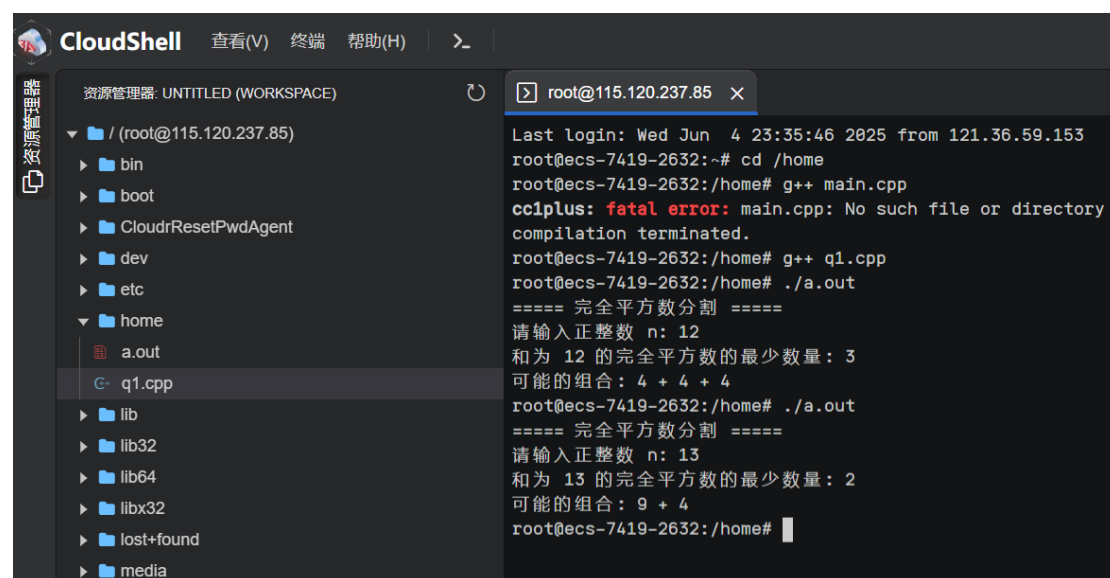
### 核心代码：

```
int numSquares(int n) {
    vector<int> dp(n+1, INT_MAX);
    dp[0] = 0;

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j*j <= i; j++) {
            dp[i] = min(dp[i], dp[i - j*j] + 1);
        }
    }
    return dp[n];
}
```

```
root@ecs-7419-2632:/home# ./a.out
===== 完全平方数分割 =====
请输入正整数 n: 12
和为 12 的完全平方数的最少数量: 3
可能的组合: 4 + 4 + 4
```

```
root@ecs-7419-2632:/home# ./a.out
===== 完全平方数分割 =====
请输入正整数 n: 13
和为 13 的完全平方数的最少数量: 2
可能的组合: 9 + 4
```



## 编程题 2：分割等和子集

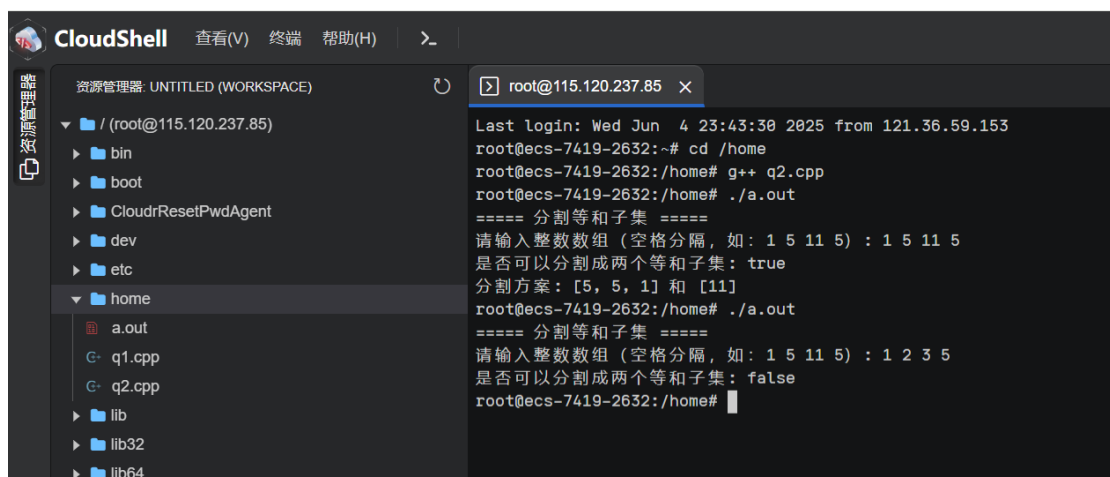
### 算法描述：

1. 计算总和，如果为奇数直接返回 false
2. 目标子集和 **target = sum/2**
3. 创建布尔数组 **dp[0..target]**，**dp[j]** 表示是否能凑出和 j
4. 初始化 **dp[0] = true**
5. 遍历每个数字：
  - 从后往前更新 **dp** 数组（避免重复使用）
  - **dp[j] = dp[j] || dp[j - num]**
6. 返回 **dp[target]**

核心代码：

```
bool canPartition(vector<int>& nums) {  
    int total = accumulate(nums.begin(), nums.end(), 0);  
    if (total % 2 != 0) return false;  
  
    int target = total / 2;  
    vector<bool> dp(target + 1, false);  
    dp[0] = true;  
  
    for (int num : nums) {  
        for (int j = target; j >= num; j--) {  
            if (dp[j - num]) {  
                dp[j] = true;  
            }  
        }  
    }  
    return dp[target];  
}
```

结果：



The screenshot shows a CloudShell terminal window with a file explorer on the left. The terminal output shows the execution of a C++ program that checks if an array can be partitioned into two equal subsets. The first test case with array [1, 5, 11, 5] returns true, and the second test case with array [1, 5, 11, 5] (likely intended to be [1, 5, 11, 5] but the output says false) returns false.

```
CloudShell 查看(V) 终端 帮助(H) >  
资源管理器: UNTITLED (WORKSPACE)  
/ (root@115.120.237.85)  
├── bin  
├── boot  
├── CloudResetPwdAgent  
├── dev  
├── etc  
└── home  
    ├── a.out  
    ├── q1.cpp  
    ├── q2.cpp  
    ├── lib  
    ├── lib32  
    └── lib64  
root@115.120.237.85 x  
Last login: Wed Jun 4 23:43:30 2025 from 121.36.59.153  
root@ecs-7419-2632:~# cd /home  
root@ecs-7419-2632:/home# g++ q2.cpp  
root@ecs-7419-2632:/home# ./a.out  
==== 分割等和子集 ====  
请输入整数数组（空格分隔，如：1 5 11 5）：1 5 11 5  
是否可以分割成两个等和子集：true  
分割方案：[5, 5, 1] 和 [11]  
root@ecs-7419-2632:/home# ./a.out  
==== 分割等和子集 ====  
请输入整数数组（空格分隔，如：1 5 11 5）：1 2 3 5  
是否可以分割成两个等和子集：false  
root@ecs-7419-2632:/home#
```