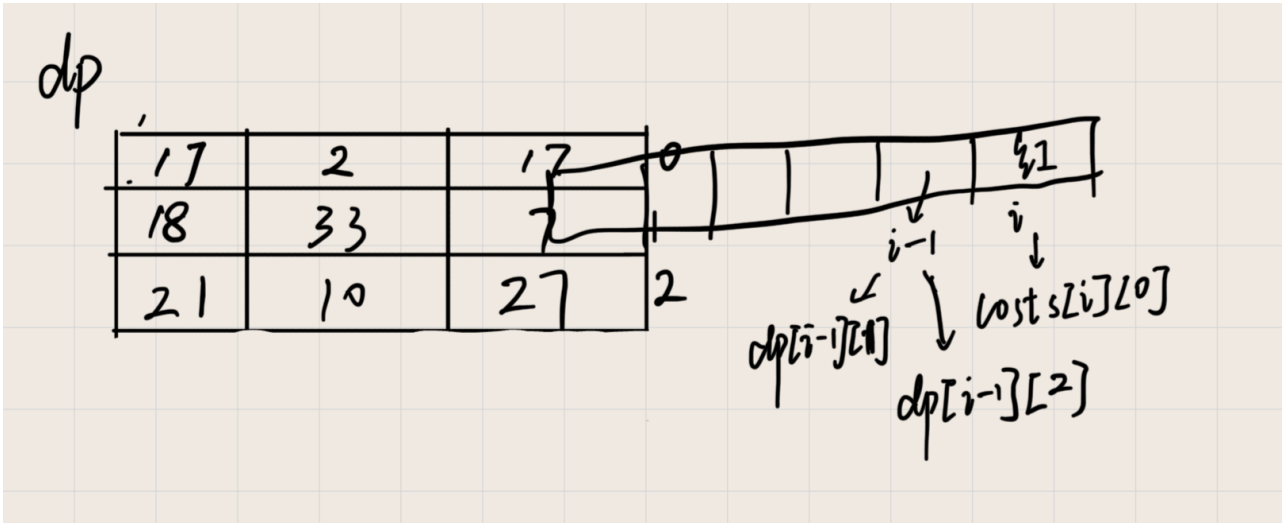
1. **动态规划：LCR 091. 粉刷房子**

解题思路：

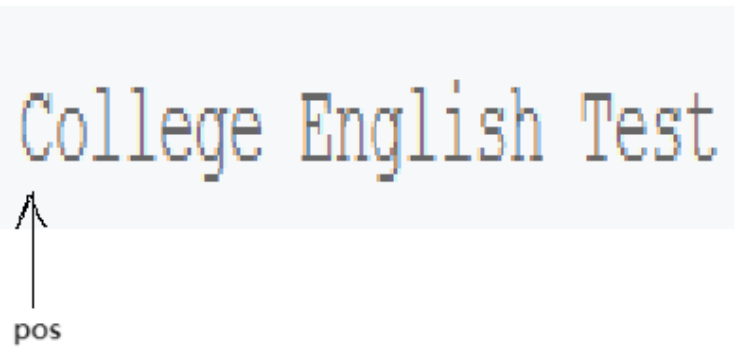


代码：

```cpp
class Solution {
public:
    int minCost(vector<vector<int>>& costs) {
        int n = costs.size();
        vector<vector<int>> dp(n,vector<int>(3));
        dp[0][0] = costs[0][0];
        dp[0][1] = costs[0][1];
        dp[0][2] = costs[0][2];

        for(int i = 1;i < n;i++)
        {
            dp[i][0] = min(dp[i-1][1],dp[i-1][2]) + costs[i][0];
            dp[i][1] = min(dp[i-1][0],dp[i-1][2])+ costs[i][1];
            dp[i][2] = min(dp[i-1][1],dp[i-1][0])+ costs[i][2];
        }
        return min(min(dp[n-1][0],dp[n-1][1]),dp[n-1][2]);

    }
};
```

2. 笔试强训：模拟 BC149 简写单词

思路：



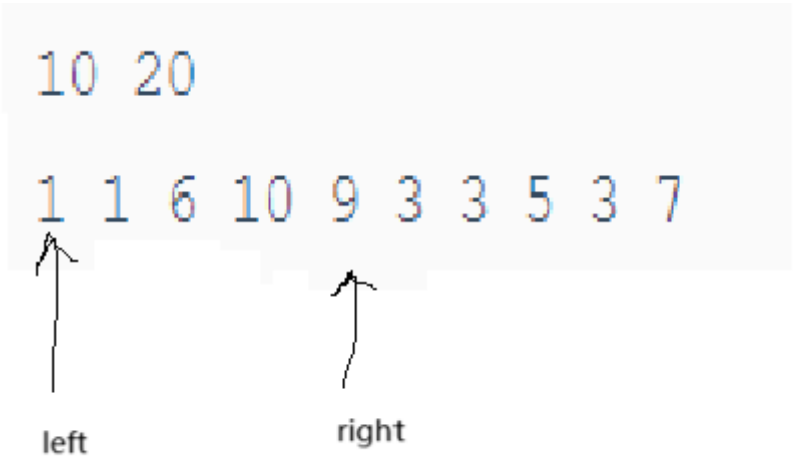
第一个特殊处理，然后一直循环直到空格下一个，如果第一个或者空格下一个是小写还要处理成大写

代码：

```cpp
#include <iostream>
#include<string>
using namespace std;

int main() {
    string in;
    string out;
    getline(cin, in);
    int n = in.size();

    size_t pos = 0;
    while (pos < n) {

        //保证pos = 0 的时候 能插入out
        if (in[pos] >= 'a' && in[pos] <= 'z') {
            out += in[pos] - 32;
        }
        else out += in[pos];
        while (pos < n && in[pos] != ' ') {
            pos++;
        }
        //保证是空格 下一个位置 并且能够保证插入完向后走
        pos++;
    }
    cout << out;
    return 0;
}
```

3. 笔试强训：滑动窗口：dd爱框框

思路：



控制一个窗口，当窗口里元素和大于特定值，更新窗口大小（看具体情况），left向右移动，直到窗口元素和小于特定值。

代码：

```cpp
#include<iostream>
#include<vector>
using namespace std;
int main()
{
    //输入
    int n = 0;
    int x = 0;
    cin >> n >>x;
    vector<long long > v(n+1);
    for(int i = 1;i <= n;i++)
    {
        cin >> v[i];
    }

    int left = 0;
    int right = 1;
    int l = 0;
    int r = 0;
    int sum = 0;
    long long len = 0x3f3f3f3f;
    while(right < n)
    {
        //进窗口
        sum += v[right];
        //判断
        while(sum >= x)
        {
            if(right - left < len)
            {
                len = right - left;
                l = left;
                r = right;
            }
            sum -= v[left++];
        }
        right++;
    }
```

```
    cout << l <<" " << r;
    return 0;
}
```