포도주 시식

https://www.acmicpc.net/problem/2156

- 포도주가 일렬로 놓여져 있고, 다음과 같은 2가지 규칙을 지키면서 포도주를 최대한 많이 마시려고 한다.
- 포도주 잔을 선택하면 그 잔에 들어있는 포도주는 모두 마셔야 하고, 마신 후에는 원래 위치에 다시 놓아야 합니다.
- 연속으로 놓여 있는 3잔을 모두 마실 수는 없다.
- dp[i][j] = i번째 포도주 잔을 마셨을 때, j번 연속으로 마신 경우의 최대양
  ① 0번 연속 : dp[i][0] = max(dp[i-1][0], dp[i-1][1], dp[i-1][2])
  ② 1번 연속 : dp[i][1] = dp[i-1][0] + A[i]
  ③ 2번 연속 : dp[i][2] = dp[i-1][1] + A[i]

```
for (int i = 1; i <= N; ++i) {
    dp[i][0] = max({ dp[i - 1][0], dp[i - 1][1], dp[i - 1][2] });
    dp[i][1] = dp[i - 1][0] + A[i];
    dp[i][2] = dp[i - 1][1] + A[i];
}

printf("%d", max({ dp[N][0], dp[N][1], dp[N][2] }));
```

포도주 시식

https://www.acmicpc.net/problem/2156

- dp[i] = A[1] ~ A[i] 까지 포도주 잔을 마셨을 때 먹을 수 있는 최대 포도주 양
  ① 0번 연속 : dp[i-1]
  ② 1번 연속 : dp[i-2] + A[i]
  ③ 2번 연속 : dp[i-3] + A[i] + A[i-1]

```
dp[1] = A[1];
dp[2] = A[1] + A[2];
for (int i = 3; i <= N; ++i) {
    dp[i] = max({ dp[i - 1], dp[i - 2] + A[i], dp[i - 3] + A[i] + A[i - 1] });
}
```