

1. 首先定義了 **Max** 函數，它接受兩個 **double** 型參數，並返回其中的最大值。
2. 然後我們定義了兩個函數，分別用於計算普通選擇權價格 (**vanilla_option**) 和下擊並且出選擇權價格 (**down_and_out_option**)。這兩個函數的結構非常相似，它們都需要以下參數：
 - **S**：標的物價格
 - **T**：到期日
 - **X**：履約價格
 - **r**：無風險利率
 - **Sigma**：標的物價格波動率
 - **n**：期數 下擊並且出選擇權函數還需要一個額外的參數 **H**：障礙價格。
3. 在這兩個函數中，首先計算出二元樹的參數，包括上升因子 **U**、下降因子 **D**、上升概率 **Pu** 和下降概率 **Pd**。然後，根據最後一期的股價計算出選擇權的 **payoff**，並儲存到一個陣列中。
4. 下擊並且出選擇權函數還需要計算障礙步數 **u_power**，如果在任何時點股價超過障礙價格，則選擇權將失效。計算 **payoff** 陣列後，會將超過障礙價格的 **payoff** 設為 0。
5. 有了最後一期的 **payoff** 之後，接著進行反向遞推計算選擇權的當期價格，這個過程也就是所謂的 "Backward Induction"。對於下擊並且出選擇權，同樣要在每一步檢查是否超過了障礙價格，如果超過就將當期價格設為 0。
6. 在 **main** 函數中，程式會提示使用者輸入各種參數，然後調用上面的函數計算普通選擇權價格和下擊並且出選擇權價格。
7. 最後，用普通選擇權價格減去下擊並且出選擇權價格，就得到了下擊並且入選擇權的價格，並將其印出。