

1.解題說明

先讓使用者輸入集合的大小 `number`，並用動態配置出一個 `number` 大的 `char` 陣列 `array`，再建立一個 `generatePowerSet` 函式傳入，`array[]`、`number`，再透過

二進制表示法去指定輸出

```
for (int i = 0; i < total; i++) { //total=總共有幾個子集合
    int subsetSize = 0;           //暫存陣列的索引
    for (int j = 0; j < n; j++) {
        if (i & (1 << j)) { //利用二進制判斷
            subset[subsetSize++] = array[j];
        }
    }
}
```

假設： $S=\{A,B,C\}$ $S=\{A, B, C\}$ $S=\{A,B,C\}$ 。

$i=1$ ：這表示你在計算第 2 個子集合（索引從 0 開始），即 $\{A\}$ 、 $\{A\}$ 、 $\{A\}$ 。

$j=0$ ：這表示你在檢查第一個元素（AAA）是否應該包含在子集合中。

位元運算： $i \& (1 \ll j) = 001 \& 001 = 001$ $i \& (1 \ll j) = 001 \& 001 = 001$ ，結果為真（即非零），表示該元素應該包含在當前子集合中。

2.效能分析

4 個元素花了 0.002 秒

10 個元素花了 0.183 秒

3.測試與驗證

4.申論及心得

```
輸入有幾個字元：4
輸入字元：A B C D
{ }
{ A }
{ B }
{ A B }
{ C }
{ A C }
{ B C }
{ A B C }
{ D }
{ A D }
{ B D }
{ A B D }
{ C D }
{ A C D }
{ B C D }
{ A B C D }
進行運算所花費的時間：0.002 秒
```

在程式中，我學習了如何使用位元運算生成集合的冪集合，這讓我對遞迴和位元運算的應用有了更深入的理解。透過動態陣列儲存子集的過程中，我意識到有效管理記憶體的重要性。同時，計時功能也讓我了解如何評估演算法的運行效率。