

主程式：

```
int main() {
    int* arr;
    int n,nb;
    cout << "請輸入陣列大小：";
    cin >> n;
    arr = new int[n];
    cout << "請輸入陣列元素：" << endl;
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    char w='Y';
    while (w=='Y') {
        cout << "請輸入還要輸入元素數量：";
        cin >> nb;
        int a = n;
        n = n + nb;
        c(arr, a, n);
        cout << "請輸入新增元素：" << endl;
        for (int i = a; i < n; i++) {
            cin >> arr[i];
        }
        cout << "是否還須要輸入新的元素？(Y/N)";
        cin >> w;
        if (w == 'N') break;
    }
    cout << "最終陣列：" << endl;
    for (int i = 0; i < n; i++) {
        cout << arr[i] << endl;
    }
    return 0;
}
```

宣告各資料功能：

arr：為陣列本體，可宣告動態陣列

n：用來初始陣列的 **size**

nb：讓使用者可輸入需要新增陣列多少大小

w：讓使用者輸入字元 **Y** or **N**，以判斷是否需要繼續擴增

a：存 **n** 原本的值

起初先 **cin>>n**，讓使用者輸入陣列大小，並且以迴圈 **cin>>arr[i]**，以定義陣列內元素。

初始化並宣告 **w='Y'**以滿足 **while** 迴圈條件，讓使用者輸入需要新增元素量

cin>>nb，且 **n=n+nb** 以代表 **new size**，呼叫函式(陣列,原 **size**,新 **size**)。

函式 **C** 修改陣列大小後，以 **for** 迴圈讓使用者輸入新的陣列大小(**i=a** 以接續原陣列最後位置)。

最後詢問是否還有需要輸入的元素，若 **w='N'**則跳出迴圈，反之則繼續迴圈。

函式 c：

```
void c(int*& arr, int size, int newsize) {  
    int* newarr = new int[newsize];  
    copy(arr, arr + size, newarr);  
    delete[] arr;  
    arr = newarr;  
}
```

函式傳入 `int*& arr` 以允許內部直接影響外部指標。

宣告 `newarr` 並以 `new int[newsize]` 使他陣列大小為新的陣列大小。

複製原陣列內容到新陣列：`copy(arr, arr+size, newarr)`，其中 `arr` 為起始位置，`arr+size` 為終點位置，`newarr` 則是複製過去的陣列。

`delete[] arr` 以釋放 `arr` 動態陣列。

將陣列 `arr` 指向 `newarr` 已完成新的動態陣列宣告。

輸出結果：

```
請輸入陣列大小：3
請輸入陣列元素：
1
2
3
請輸入還要輸入元素數量：3
請輸入新增元素：
6
2
4
是否還須要輸入新的元素？(Y/N)Y
請輸入還要輸入元素數量：4
請輸入新增元素：
2
5
1
2
是否還須要輸入新的元素？(Y/N)Y
請輸入還要輸入元素數量：4
請輸入新增元素：
5
6
4
8
是否還須要輸入新的元素？(Y/N)N
1
2
3
6
2
4
2
5
1
2
5
6
4
8
```