姓名： 朱育民
學號： 407410014
email： samchu0218.momo@gmail.com

使用語言： c++
版本： c++17

操作說明：
輸入選項 1~8 分別代表以下八種基本功能，在一開始執行程式時，一定得先選擇選項 1（Initialize）來建立一棵 B+ Tree，或是輸入 8 離開程式。
每做完一個動作，都會顯示選單供使用者選擇欲執行項目。

(1) Initialize:
輸入一個正整數 initializing order，來代表想建立 B+ Tree 的 order，輸入完後按下 Enter 即可成功建立該 B+ Tree。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 1
Initializing order = 2
Success create a B+ tree with order 2
```

(2) Attach
會先要求輸入欲使用的 B+ Tree order，為一個正整數。
再輸入欲建立的節點資料(必須為合法的 B+ Tree)，輸入完後按下 Enter 即可建立新的 B+ Tree 並拿來使用，而舊的樹會被刪除。
(輸入格式為 hw-5.pdf p.12-13，leaf node 中資料以空白隔開，node 間則以" ；"隔開，且分號與數字間也會以空白隔開，並以中序排列)

輸入範例: 3 4 5 6 ; 7 ; 8 10 17 19 ; 25 ; 28 33 ; 35 ; 42 51

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 2
Attaching order = 2
Nodes in inorder-like traversal = 3 4 5 6 ; 7 ; 8 10 17 19 ; 25 ; 28 33 ; 35 ; 42 51
Remove B+ Tree Success!
Attaching Success! create a B+ tree with order 2
```

(3) Bulkload

只能對一棵空的 B+ Tree 做此操作，輸入一串用來建立 B+ Tree 的正整數，並以空白隔開，按下 Enter 即可以 Bulkload 的方式建立此 B+ Tree。

輸入範例: 46 10 70 49 23 40 59 29 34 54 75 30

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 3
Bulkloading key sequence: 46 10 70 49 23 40 59 29 34 54 75 30
Try to Insert value: 10
Try to Insert value: 23
Try to Insert value: 29
Try to Insert value: 30
Try to Insert value: 34
Try to Insert value: 40
Try to Insert value: 46
Try to Insert value: 49
Try to Insert value: 54
Try to Insert value: 59
Try to Insert value: 70
Try to Insert value: 75
BulkLoading Success!
```

(4) Lookup

輸入一個正整數，若該數存在於目前的 B+ Tree 中，則顯示 true，若不存在則顯示 false。

以 Bulkload 輸入範例建立的 B+ Tree 為例

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 4
Look up key = 10

true

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 4
Look up key = 55

false

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 4
Look up key = 34

true
```

(5) Insert

輸入一個正整數，按下 Enter 後，會將該數加進目前的 B+ Tree 中。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 5
Insert key = 11
Try to Insert value: 11
Insert Success !

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 5
Insert key = 94
Try to Insert value: 94
Insert Success !

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 5
Insert key = 37
Try to Insert value: 37
Insert Success !
```

(6) Delete

輸入一個正整數，按下 Enter 後，會移除目前 B+ Tree 中的該正整數。若該正整數不存在則回報錯誤。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 6
Delete key = 94
Try to Delete value: 94
Delete Success !

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 6
Delete key = 10
Try to Delete value: 10
Error! Key: 10 Not Exist
Delete Fail !
```

(7) Display

顯示目前的 B+ Tree，若該樹為空，則顯示 Empty Tree。

除了以 attach 資料輸入格式來輸出之外，也會在終端機上顯示該 B+ Tree 大概的架構，大括號內為 leaf node，中括號內為 internal node，而最上層為 root，最下層為 leaf ndoe，一個 internal node 的兩端則分別代表指向子節點的 pointer。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 1
Initializing order = 2
Success create a B+ tree with order 2

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7
Empty Tree
```

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 2
Attaching order = 2
Nodes in inorder-like traversal = 3 4 5 6 ; 7 ; 8 10 17 19 ; 25 ; 28 33 ; 35 ; 42 51
Remove B+ Tree Success!
Attaching Success! create a B+ tree with order 2

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

         [ 7              25          35 ]

{ 3 4 5 6 }   { 8 10 17 19 }    { 28 33 }    { 42 51 }


Nodes in inorder-like traversal:
3 4 5 6 ; 7 ; 8 10 17 19 ; 25 ; 28 33 ; 35 ; 42 51
```

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 1
Initializing order = 1
Success create a B+ tree with order 1

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 3
Bulkloading key sequence: 46 10 70 49 23 40 59 29 34 54 75 30 7 2 9 87 55 31 20
Try to Insert value: 2
Try to Insert value: 7
Try to Insert value: 9
Try to Insert value: 10
Try to Insert value: 20
Try to Insert value: 23
Try to Insert value: 29
Try to Insert value: 30
Try to Insert value: 31
Try to Insert value: 34
Try to Insert value: 40
Try to Insert value: 46
Try to Insert value: 49
Try to Insert value: 54
Try to Insert value: 55
Try to Insert value: 59
Try to Insert value: 70
Try to Insert value: 75
Try to Insert value: 87
BulkLoading Success!

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

                                                  [ 49 ]

                        [ 29 ]                                          [ 70 ]

     [ 9         20 ]               [ 31          40 ]               [ 55 ]               [ 87 ]

{ 2 7 }   { 9 10 }    { 20 23 }    { 29 30 }    { 31 34 }    { 40 46 }    { 49 54 }    { 55 59 }    { 70 75 }    { 87 }


Nodes in inorder-like traversal:
2 7 ; 9 ; 9 10 ; 20 ; 20 23 ; 29 ; 29 30 ; 31 ; 31 34 ; 40 ; 40 46 ; 49 ; 49 54 ; 55 ; 55 59 ; 70 ; 70 75 ; 87 ; 87
```

(8) Quit

結束程式

B+ Tree key 值限制:

無法支援非正整數的輸入。

無法支援一棵樹中存在相同的資料。

非自行完成的部分:

無

Bonus:

(1)實作出 insert 做出 re-distribution

leaf page overflow 時，能夠支援 re-distribution，index page overflow 時，也能支援 non-leaf 的 re-distribution。

若發生 overflow 的節點，其左節點(left sibling)與右節點(right sibling)都沒有足夠的空間做 re-distribution 時，才會執行 page splitting。

範例: B+ Tree order = 2，

加入 key = 45，左右都無空間，只能做 page splitting。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

              [ 34                54 ]

{ 10 23 29 30 }    { 34 40 46 49 }    { 54 59 70 75 }


Nodes in inorder-like traversal:
10 23 29 30 ; 34 ; 34 40 46 49 ; 54 ; 54 59 70 75

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 5
Insert key = 45
Try to Insert value: 45
Insert Success !

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

              [ 34          45             54 ]

{ 10 23 29 30 }    { 34 40 }    { 45 46 49 }    { 54 59 70 75 }


Nodes in inorder-like traversal:
10 23 29 30 ; 34 ; 34 40 ; 45 ; 45 46 49 ; 54 ; 54 59 70 75
```

範例: B+ Tree order = 2，
加入 key = 20，原本節點產生 overflow，但右節點仍有空間，因此可以做 re-distribution。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

             [ 34           45              54 ]

{ 10 23 29 30 }    { 34 40 }    { 45 46 49 }    { 54 59 70 75 }


Nodes in inorder-like traversal:
10 23 29 30 ; 34 ; 34 40 ; 45 ; 45 46 49 ; 54 ; 54 59 70 75

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 5
Insert key = 20
Try to Insert value: 20
Insert Success !

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

             [ 30              45              54 ]

{ 10 20 23 29 }    { 30 34 40 }    { 45 46 49 }    { 54 59 70 75 }


Nodes in inorder-like traversal:
10 20 23 29 ; 30 ; 30 34 40 ; 45 ; 45 46 49 ; 54 ; 54 59 70 75
```

範例: B+ Tree order = 1，
加入 key = 99，因為 leaf page 產生 overflow，在做完 page splitting 後，換上層的 index page 產生 overflow，但因為其左邊的節點還有空間，因此會做 non-leaf 的 re-distribution。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

                            [ 40                        54 ]

       [ 29          34 ]                  [ 46 ]                  [ 70          72 ]

{ 10 23 }    { 29 30 }    { 34 36 }    { 40 45 }    { 46 49 }    { 54 59 }    { 70 71 }    { 72 75 }


Nodes in inorder-like traversal:
10 23 ; 29 ; 29 30 ; 34 ; 34 36 ; 40 ; 40 45 ; 46 ; 46 49 ; 54 ; 54 59 ; 70 ; 70 71 ; 72 ; 72 75

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 5
Insert key = 99
Try to Insert value: 99
Insert Success !

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

                            [ 40                                    70 ]

       [ 29          34 ]                  [ 46          54 ]                  [ 72        75 ]

{ 10 23 }    { 29 30 }    { 34 36 }    { 40 45 }    { 46 49 }    { 54 59 }    { 70 71 }    { 72 }    { 75 99 }


Nodes in inorder-like traversal:
10 23 ; 29 ; 29 30 ; 34 ; 34 36 ; 40 ; 40 45 ; 46 ; 46 49 ; 54 ; 54 59 ; 70 ; 70 71 ; 72 ; 72 ; 75 ; 75 99
```

(2)實作出 Delete 的 re-distribution & merging

leaf page 與 index page 在 underflow 發生時，都可以支援 re-distribution (non-leaf re-distribution) 與 merging。

當一個節點發生 underflow 時，會先判斷是否可以使用左節點(left sibling)或是右節點(right sibling)的 key 做 re-distribution (non-leaf re-distribution)，若都不行的話，則再決定要向左節點或右節點做 merging。若 underflow 的節點為 root(容納的 key 數量為 0)時，則刪除此 root page，並將 root 指標指向原本 root page 的唯一子節點。

範例: B+ Tree order = 2，
移除 key = 10，leaf page 發生 underflow，因為可以向右節點拿一個 key，因此會做 re-distribution。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

                                  [ 42 ]

       [ 24             34 ]                      [ 54           70 ]

{ 10 23 }    { 24 29 30 }    { 34 40 }    { 42 46 49 }    { 54 59 }    { 70 75 80 }


Nodes in inorder-like traversal:
10 23 ; 24 ; 24 29 30 ; 34 ; 34 40 ; 42 ; 42 46 49 ; 54 ; 54 59 ; 70 ; 70 75 80

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 6
Delete key = 10
Try to Delete value: 10
Delete Success !

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

                                [ 42 ]

       [ 29           34 ]                      [ 54           70 ]

{ 23 24 }    { 29 30 }    { 34 40 }    { 42 46 49 }    { 54 59 }    { 70 75 80 }


Nodes in inorder-like traversal:
23 24 ; 29 ; 29 30 ; 34 ; 34 40 ; 42 ; 42 46 49 ; 54 ; 54 59 ; 70 ; 70 75 80
```

範例: B+ Tree order = 2，
移除 key = 40，leaf page 發生 underflow，但無法向左右節點拿 key，因此只能向左做 merging，而後會導致上層的 index page 產生 underflow，又會因為左右 sibling 都沒有多的 key 可以拿來用，因此會再次做 merging。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

                                [ 42 ]

        [ 29           34 ]                      [ 54           70 ]

{ 23 24 }    { 29 30 }    { 34 40 }    { 42 46 49 }    { 54 59 }    { 70 75 80 }


Nodes in inorder-like traversal:
23 24 ; 29 ; 29 30 ; 34 ; 34 40 ; 42 ; 42 46 49 ; 54 ; 54 59 ; 70 ; 70 75 80

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 6
Delete key = 40
Try to Delete value: 40
Removing leaf page {  }
Delete Success !

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

        [ 29             42             54           70 ]

{ 23 24 }    { 29 30 34 }    { 42 46 49 }    { 54 59 }    { 70 75 80 }


Nodes in inorder-like traversal:
23 24 ; 29 ; 29 30 34 ; 42 ; 42 46 49 ; 54 ; 54 59 ; 70 ; 70 75 80
```

範例: B+ Tree order = 2，

移除 key = 29，leaf page 發生 underflow，但無法向左右節點拿 key，因此只能先向右做 merging，做完後會導致上層的 index page 產生 underflow，但其右邊節點有多的 key 可以拿來做 non-leaf 的 re-distribution。

```
(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

                              [ 42 ]

       [ 24          30 ]                     [ 54               80          91 ]

{ 10 23 }    { 24 29 }    { 30 34 }    { 42 46 49 }    { 54 59 70 75 }    { 80 85 }    { 91 99 101 }


Nodes in inorder-like traversal:
10 23 ; 24 ; 24 29 ; 30 ; 30 34 ; 42 ; 42 46 49 ; 54 ; 54 59 70 75 ; 80 ; 80 85 ; 91 ; 91 99 101

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 6
Delete key = 29
Try to Delete value: 29
Removing leaf page {  }
Delete Success !

(1) Initialize (2) Attach (3) Bulkload (4) Lookup
(5) Insert     (6) Delete (7) Display  (8) Quit
Select an operation: 7

                                  [ 54 ]

       [ 24            42 ]                          [ 80          91 ]

{ 10 23 }    { 24 30 34 }    { 42 46 49 }    { 54 59 70 75 }    { 80 85 }    { 91 99 101 }


Nodes in inorder-like traversal:
10 23 ; 24 ; 24 30 34 ; 42 ; 42 46 49 ; 54 ; 54 59 70 75 ; 80 ; 80 85 ; 91 ; 91 99 101
```

螢幕錄製 demo 影片：

螢幕錄影 1:
示範 Initialize、Bulkload、Insert、Display 功能。

螢幕錄影 2:
示範 Attach、Lookup、Delete、Display、Quit 功能。

螢幕錄影 3:
示範 InitializeAttach、Insert、Delete、Display、Quit 功能。

螢幕錄影 4、螢幕錄影 5:
示範 Insert 可以根據左右 sibling 的 key 數量，決定執行 page splitting、re-distribution 或是 non-leaf 的 re-distribution。

螢幕錄影 6、螢幕錄影 7:
示範 Delete 可以根據左右 sibling 的 key 數量，決定執行 merging、re-distribution 或是 non-leaf 的 re-distribution。