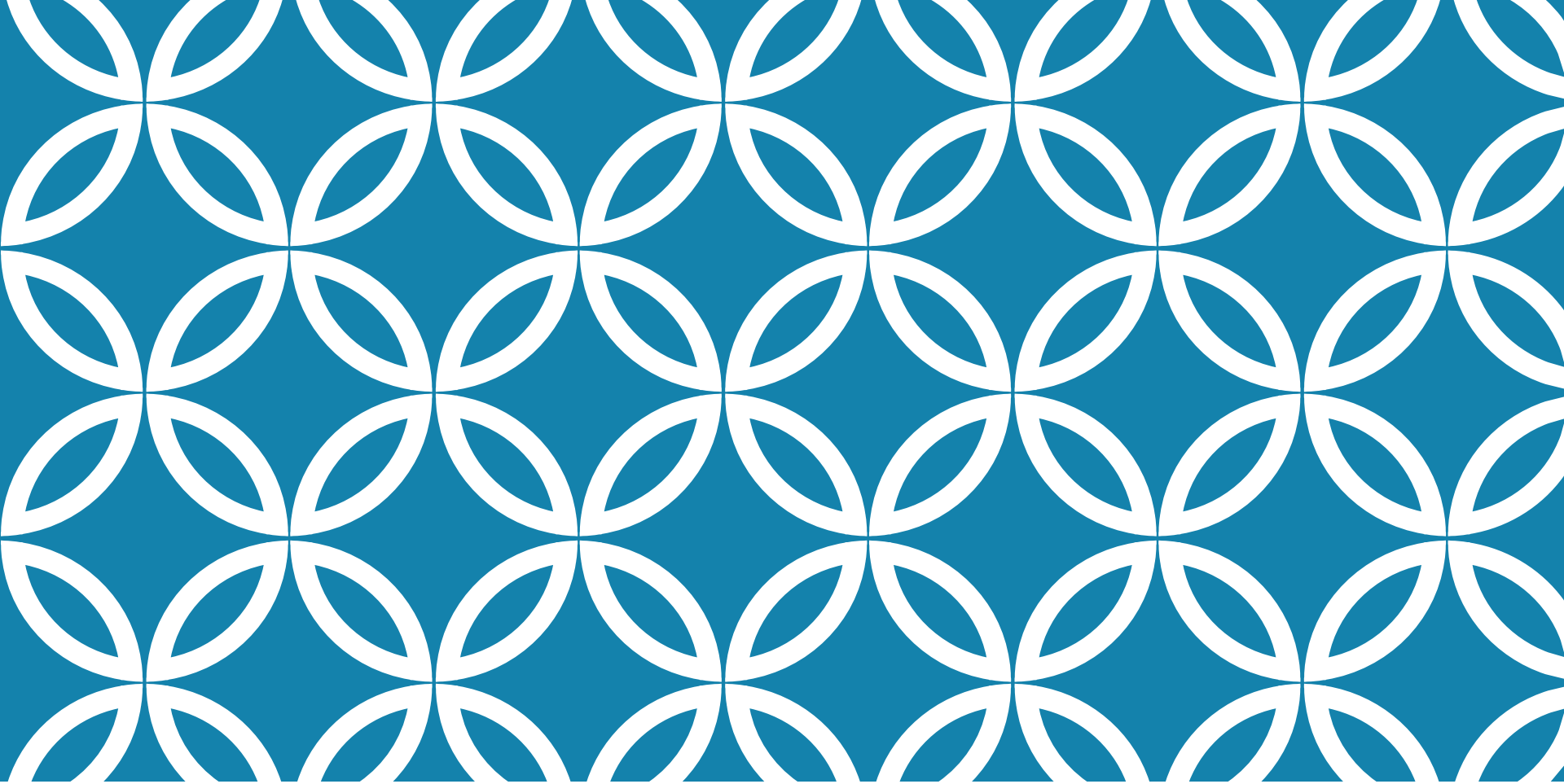
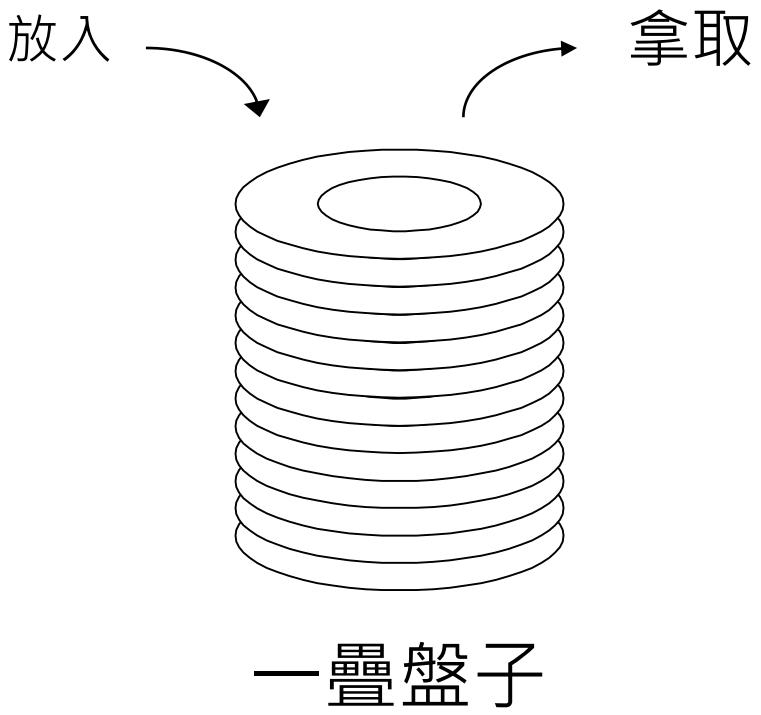


第三章 堆疊

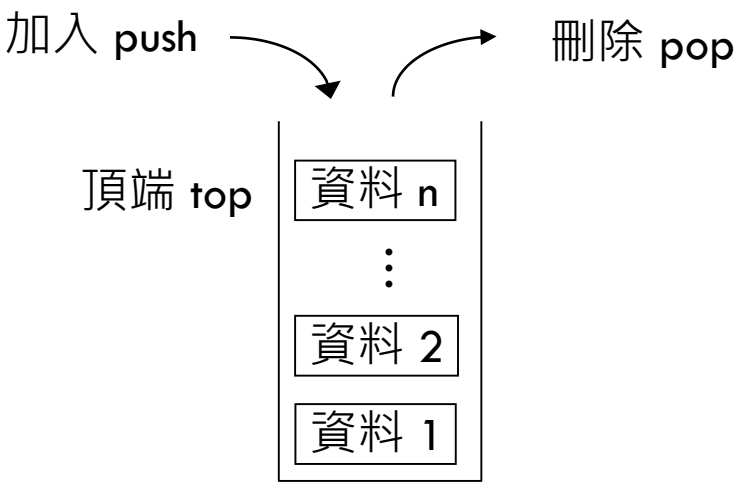
資料結構
鍾宜玲

你一定玩過堆疊的遊戲！



堆疊(STACKS)的定義

- 一個 **後進先出 (LIFO, Last In First Out)** 的有序串列。
- 資料的加入與刪除僅在串列的 **頂端 (top)**。
- 加入資料於堆疊內通常稱為 **推入push**。
- 刪除堆疊內的資料一般稱為 **彈出pop**。

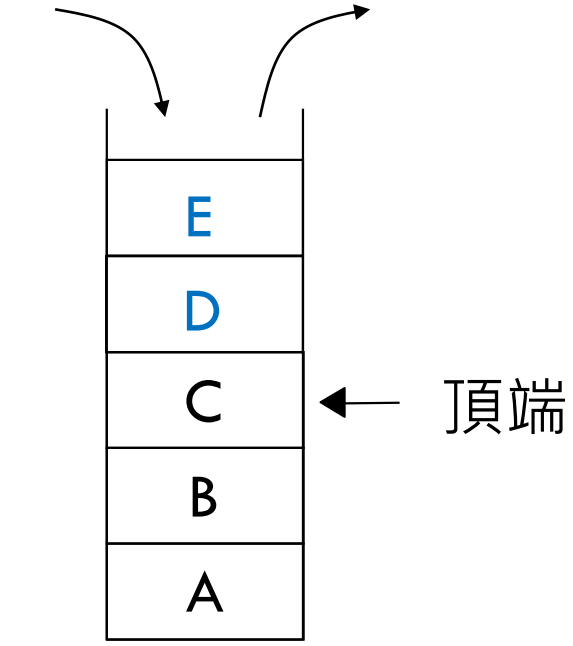


堆疊的運作

(1) 加入資料 D

(2) 加入資料 E

(3) 刪除，傳回 E



堆疊 = (A, B, C, D, E)

以陣列製作堆疊

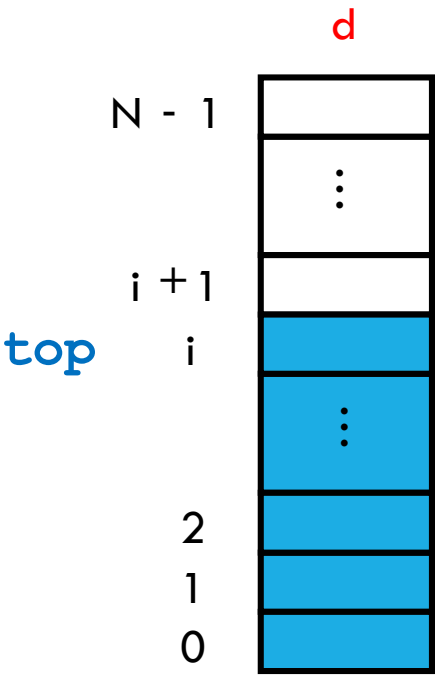
- 若堆疊的資料是整數，堆疊的最大容量是100

```
#define N 100     //堆疊大小
int stack[N];     //陣列stack當作堆疊
int top = -1;     //top表頂端的位置(陣列索引)
```

- 初始堆疊為空的，故 top 之初始值設為 -1。

加入資料 PUSH()

加入一筆資料d，則 top 值增加1。




```
top++;
stack[top]=d;
```

↓

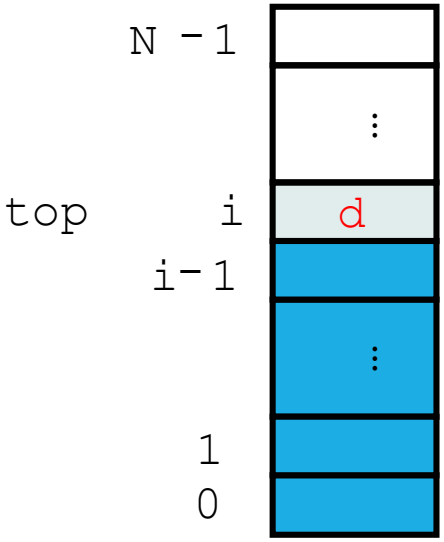
```
stack[++top]=d;
```

加入資料於陣列堆疊(續)

```
void push(int  d)
{
  if( top == N-1 ){          //檢查堆疊是否滿了，
     printf("堆疊滿了\n");    //若是滿了則加入失敗。
     exit(1);                //加入失敗，結束程式執行
  }
  stack[++top]=d;            //否則將top加1，新資料加入陣
                             列中(索引top)。

}
```

刪除堆疊頂端資料



`return stack[top--];`

傳回頂端資料，並將top值減1。

(頂端內容不必清除)

取出堆疊頂端資料_程式

```
int pop()
{
  if(top == -1){                    //檢查堆疊是否空了
     printf("堆疊空了\n");          //若是空堆疊則刪除失敗
     exit(1);                       //刪除失敗，結束執行
  }
 return(stack[top--]);              //傳回頂端資料，top值減一
}
```

例.

假設堆疊的大小是6，push ( ) 函數會將資料加入堆疊內，pop ( ) 函數會取出堆疊頂端資料。若主程式如下，請寫出printf ( ) 執行的結果。

```
void main()
{
  push(10);
  push(20);
  push(30);
  push( pop() + 40);
  printf("%d\n", pop());
  printf("%d\n", pop()-pop());
}
```

執行結果：

70

10