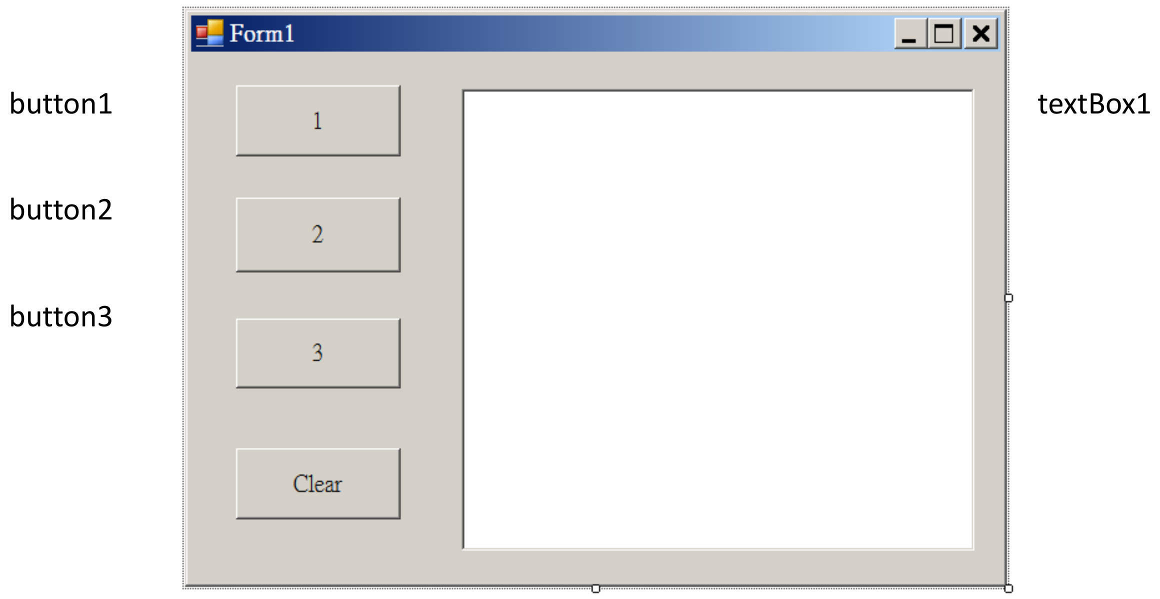
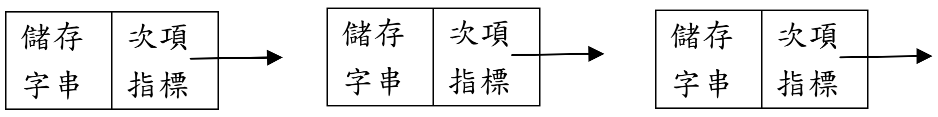
Ex. 1 練習使用串列儲存資料.



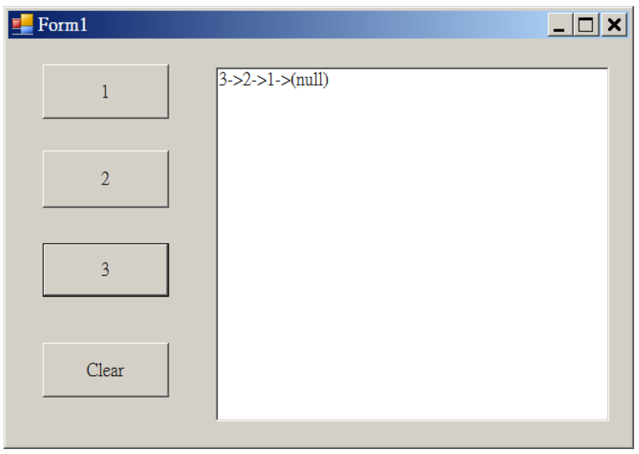
(資料結構)

串列節點 node



```
class list {
        String  ^name; //儲存字串
        list    ^next; //次項指標
} ^A;
```

開始時 A 指向無效指標(null), 代表空串列
(null)



```
3->2->1->(null)
```

加入 "1"
"1" -> (null)

加入 "2"
"2" -> "1" -> (null)

加入 "3"
"3" -> "2" -> "1" -> (null)

(程式碼)

```
      ref class list {
       public:
          String ^name;
          list   ^next;
          list(){ name=""; next=nullptr; }
      } ^A;
      void add(String ^s){
          list ^x = gcnew list;
          x->next = A;
          x->name = s;
          A=x;
      }
      void show(){
          textBox1->Text="";
          list ^x=A;
          while( x !=nullptr ){
              textBox1->Text += x->name + "->";
              x=x->next;
          }
          textBox1->Text += "(null)";
      }
  private: System::Void Form1_Load(System::Object^
                              sender, System::EventArgs^  e) {
              A=nullptr;
              show();
      }
  private: System::Void button1_Click(System::Object^
                              sender, System::EventArgs^  e) {
              add( button1->Text );
              show();
      }
  private: System::Void button2_Click(System::Object^
                              sender, System::EventArgs^  e) {
              add( button2->Text );
              show();
       }
```

```
private: System::Void button3_Click(System::Object^
                         sender, System::EventArgs^  e) {
          add( button3->Text );
          show();
     }
private: System::Void button4_Click(System::Object^
                         sender, System::EventArgs^  e) {
           A=nullptr;
           show();
     }
```

Ex. 2 同 Ex. 1使用包裝(Encapsulation)的概念來設計串列類別.

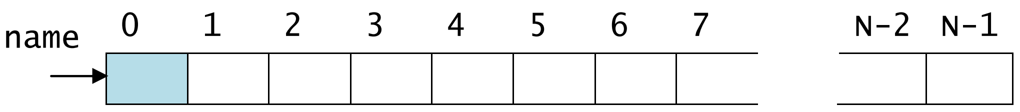
```
ref class list {
 public:
    String ^name;
    list   ^next;
    list( String ^s ){
        name=s;
        next=nullptr;
    }
};
ref class LIST {
  private:
    list ^L;
  public:
    LIST() { L=nullptr; }
    void add( String ^s ){
        list ^x = gcnew list( s );
        x->next = L;
        L=x;
    }
    String^ show(){
        String ^s = "";

        list ^x=L;
        while( x !=nullptr ){
            s += x->name + "->";
            x=x->next;
        }
        s += "(null)";
        return s;
    }
    void clear(){ L=nullptr; }
};
```

```
LIST ^A;
private: System::Void Form1_Load(System::Object^
                     sender, System::EventArgs^  e) {
           A=gcnew LIST;
           textBox1->Text = A->show();
    }
private: System::Void button1_Click(System::Object^
                     sender, System::EventArgs^  e) {
           A->add( button1->Text );
           textBox1->Text = A->show();
    }
private: System::Void button2_Click(System::Object^
                     sender, System::EventArgs^  e) {
           A->add( button2->Text );
           textBox1->Text = A->show();
     }
private: System::Void button3_Click(System::Object^
                     sender, System::EventArgs^  e) {
           A->add( button3->Text );
           textBox1->Text = A->show();
     }
private: System::Void button4_Click(System::Object^
                     sender, System::EventArgs^  e) {
           A->clear();
           textBox1->Text = A->show();
     }
```

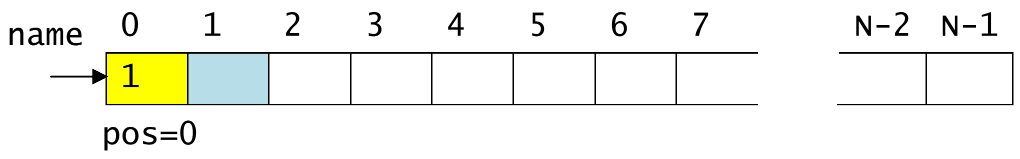
Ex. 3 同 Ex. 1 使用包裝概念設計陣列類別來儲存.

資料結構
(初始)

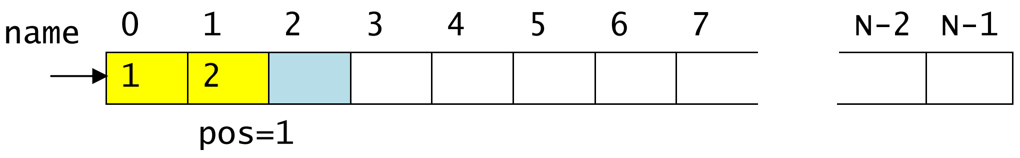


陣列最後元素索引位置 pos, 可填入索引位置 pos+1
pos = -1 表示空陣列

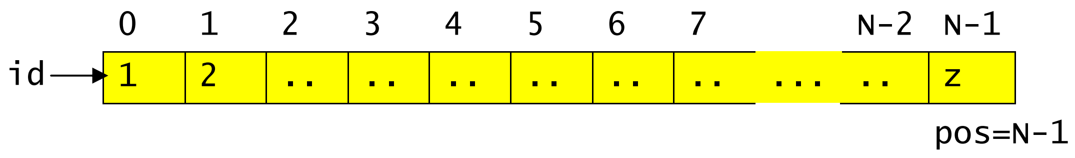
(新增 "1")

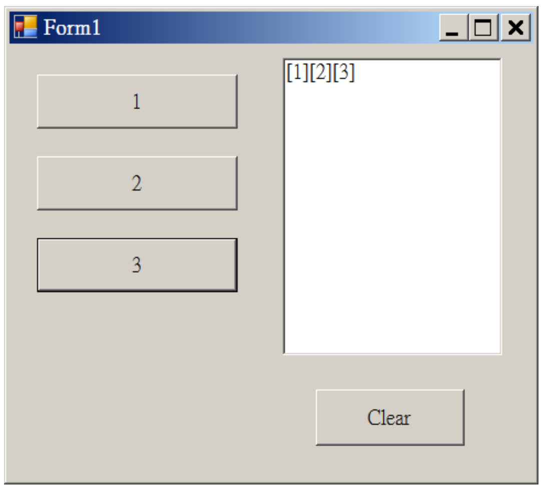


(新增 "2")



(陣列全滿的情況)





開始時 A 是空陣列, pos = -1

加入 "1", pos = 0
[1]

加入 "2", pos = 1
[1][2]

加入 "3", pos =2
[1][2][3]

```
ref class ARRAY{
 private:
    int N;
    array<String^> ^name;
    int pos;
 public:
    ARRAY(){
        N=5;
        pos=-1;
        name = gcnew array<String^>( N );
    }

    void add( String ^x ){
        if( pos >= N-1) {
             N = 2*N;
             array<String^> ^s
                    = gcnew array<String^>( N );
             int k;
             for(k=0; k< N/2; k++) s[k]=name[k];
             name = s;
        }
        ++pos;
        name[pos]= x;
     }
    String^ show(){
        String ^s="";
        int k;
        for(k=0; k<= pos; k++ )
            s += "[" + name[k] + "]";
        return s;
    }
    void clear(){
         pos=-1;
    }
 };
```

```
ARRAY ^A;
private: System::Void Form1_Load(System::Object^
                    sender, System::EventArgs^  e) {
    A=gcnew ARRAY;
    textBox1->Text = A->show();
 }
private: System::Void button1_Click(System::Object^
                    sender, System::EventArgs^  e) {
    A->add( button1->Text );
    textBox1->Text = A->show();
 }
private: System::Void button2_Click(System::Object^
                    sender, System::EventArgs^  e) {
    A->add( button2->Text );
    textBox1->Text = A->show();
 }
private: System::Void button3_Click(System::Object^
                    sender, System::EventArgs^  e) {
    A->add( button3->Text );
    textBox1->Text = A->show();
 }
 private: System::Void button4_Click(System::Object^
                    sender, System::EventArgs^  e) {
    A->clear();
    textBox1->Text = A->show();
 }
```