Chapter 3

COLLECTIONS

COLLECTIONS

Overview

- COLLECTIONS เป็นการศึกษาเกี่ยวกับ Class ของ Collection ซึ่งสามารถจัดเก็บข้อมูลลงไปในตัวแปรได้หลายค่า และสามารถดึงข้อมูลเหล่านั้นกลับออกมาใช้งานได้
- Collection มี Class ย่อย ๆ ประกอบไปด้วย
  - Lists
  - Hash tables
  - Stacks
  - Queues

Collection Classes and Their Usage

Class	Description and Useage
ArrayList	เป็นการจัดเก็บ Object แบบเรียงลำดับโดยมี Index ที่ไม่ซ้ำกันกำกับข้อมูลอยู่
Hashtable	ใช้ Key ในการเข้าถึงข้อมูลที่เก็บไว้ใน Collection
Stack	คือโครงสร้าง last-in, first-out ในการเก็บข้อมูล
Queue	คือโครงสร้าง first-in, first-out ในการเก็บข้อมูล

ArrayList Class

- เป็นการจัดเก็บ Object แบบเรียงลำดับโดยมี Index ที่ไม่ซ้ำกันกำกับข้อมูลอยู่
- มีขนาดจำกัด ใช้สำหรับรองรับข้อมูลที่ไม่แน่ชัด
- ArrayList คือโครงสร้างข้อมูลแบบ Linked-List นั่นเอง
- ArrayList ไม่เหมือน Array
  - สามารถเพิ่ม หรือ ลบข้อมูลตามตำแหน่งที่ต้องการได้
  - ArrayList สามารถปรับขนาดของตัวเองได้อัตโนมัติ
  - ช่วยจัดสรรหน่วยความจำในแบบ Dynamic ช่วยในการ Adding, Searching และ Sorting ใน ArrayList ได้

Some Properties of ArrayList Class

Property	Description
Capacity	Gets or sets the number of elements that the ArrayList can contain.
Count	Gets the number of elements actually contained in the ArrayList.
IsFixedSize	Gets a value indicating whether the ArrayList has a fixed size.
IsReadOnly	Gets a value indicating whether the ArrayList is read-only.
Item	Gets or sets the element at the specified index.

Some Methods of ArrayList Class

No.	Methods
1	**public virtual int Add(object value);** Adds an object to the end of the ArrayList.
2	**public virtual void AddRange(ICollection c);** Adds the elements of an ICollection to the end of the ArrayList.
3	**public virtual void Clear();** Removes all elements from the ArrayList.
4	**public virtual bool Contains(object item);** Determines whether an element is in the ArrayList.
5	**public virtual ArrayList GetRange(int index, int count);** Returns an ArrayList which represents a subset of the elements in the source ArrayList.
6	**public virtual int IndexOf(object);** Returns the zero-based index of the first occurrence of a value in the ArrayList or in a portion of it.
7	**public virtual void Insert(int index, object value);** Inserts an element into the ArrayList at the specified index.
8	**public virtual void InsertRange(int index, ICollection c);** Inserts the elements of a collection into the ArrayList at the specified index.

No.	Methods
9	**public virtual void Remove(object obj);** Removes the first occurrence of a specific object from the ArrayList.
10	**public virtual void RemoveAt(int index);** Removes the element at the specified index of the ArrayList.
11	**public virtual void RemoveRange(int index, int count);** Removes a range of elements from the ArrayList.
12	**public virtual void Reverse();** Reverses the order of the elements in the ArrayList.
13	**public virtual void SetRange(int index, ICollection c);** Copies the elements of a collection over a range of elements in the ArrayList.
14	**public virtual void Sort();** Sorts the elements in the ArrayList.
15	**public virtual void TrimToSize();** Sets the capacity to the actual number of elements in the ArrayList.

ArrayList Class Example


```
using System;
using System.Collections;

namespace Chapter4
{
    class Program
    {
        static void Main(string[] args)
        {
            ArrayList al = new ArrayList();
            Console.WriteLine("Adding some numbers:");
            al.Add(45);
            al.Add(78);
            al.Add(33);
            al.Add(56);
            al.Add(12);
            al.Add(23);
            al.Add(9);
            Console.WriteLine("Capacity: " + al.Capacity);
            Console.WriteLine("Count: " + al.Count);
            Console.ReadLine();
        }
    }
}
```

OUTPUT

```
Adding some numbers:
Capacity: 8
Count: 7
```

```
using System;
using System.Collections;

namespace Chapter4
{
    class Program
    {
        static void Main(string[] args)
        {
            ArrayList al = new ArrayList();
            Console.WriteLine("Adding some numbers:");
            al.Add(45);
            al.Add(78);
            al.Add(33);
            al.Add(56);
            al.Add(12);
            al.Add(23);
            al.Add(9);
            Console.WriteLine("Capacity: " + al.Capacity);
            Console.WriteLine("Count: " + al.Count);
            Console.Write("Content: ");
            foreach (int i in al)
            {
                Console.Write(i + " ");
            }
            Console.WriteLine();
            Console.Write("Sorted Content: ");
            al.Sort();
            foreach (int i in al)
            {
                Console.Write(i + " ");
            }
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}
```

ArrayList Class Example


```
for (int i = 0; i <= al.Count - 1; i++)
{
    Console.Write(al[i] + " ");
}
```

OUTPUT

```
Adding some numbers:
Capacity: 8
Count: 7
Content: 45 78 33 56 12 23 9
Content: 9 12 23 33 45 56 78
```

```
using System;
using System.Collections;

namespace Chapter4
{
    class Program
    {
        static void Main(string[] args)
        {
            ArrayList al = new ArrayList();
            al.Add(1);
            al.Add(2);
            al.Add(3);
            al.Add(5);
            al.Add(6);
            al.Add(7);
            al.Add(8);
            Console.Write("Content: ");
            foreach (int i in al)
            {
                Console.Write(i + " ");
            }
            Console.WriteLine();
            al.Insert(3, 4);
            Console.Write("After Insert: ");
            foreach (int i in al)
            {
                Console.Write(i + " ");
            }
            Console.WriteLine();
            Console.ReadKey();
        }
    }
}
```

ArrayList Class Example

> Insert

OUTPUT

```
Content: 1 2 3 5 6 7 8
After Insert: 1 2 3 4 5 6 7 8
```

```
using System;
using System.Collections;

namespace Chapter4
{
    class Program
    {
        static void Main(string[] args)
        {
            ArrayList al = new ArrayList();
            int find = 7;
            al.Add(1);
            al.Add(2);
            al.Add(3);
            al.Add(5);
            al.Add(6);
            al.Add(7);
            al.Add(8);
            Console.Write("Content: ");
            foreach (int i in al)
            {
                Console.Write(i + " ");
            }
            Console.WriteLine();
            int f = al.IndexOf(find);
            if (f >= 0)
            {
                Console.WriteLine("Found: " + al[f] +
                    " at Position: " + f);
            }
            else
            {
                Console.WriteLine(find + " Not Found");
            }
            Console.ReadKey();
        }
    }
}
```

ArrayList Class Example

> Find

OUTPUT

```
Content: 1 2 3 5 6 7 8
Found: 7 at Position: 5
```

Hashtable Class

- ใช้ Key ในการเข้าถึงข้อมูลที่เก็บไว้ใน Collection
- Hashtable เป็นโครงสร้างข้อมูลที่ใช้สำหรับเก็บข้อมูลเพื่อการสืบค้นที่รวดเร็ว แต่แก้ไขได้ช้า เหมาะกับข้อมูลที่ถูกใช้บ่อยแต่ไม่ค่อยได้แก้ไข
- ข้อมูลใน Hashtable จะสามารถ Add Object ได้ 2 ค่า ประกอบด้วย Key และ Values

Some Properties of Hashtable Class

Property	Description
Count	Gets the number of key-and-value pairs contained in the Hashtable.
IsFixedSize	Gets a value indicating whether the Hashtable has a fixed size.
IsReadOnly	Gets a value indicating whether the Hashtable is read-only.
Item	Gets or sets the value associated with the specified key.
Keys	Gets an ICollection containing the keys in the Hashtable.
Values	Gets an ICollection containing the values in the Hashtable.

Some Methods of Hashtable Class

No.	Method
1	**public virtual void Add(object key, object value);** Adds an element with the specified key and value into the Hashtable.
2	**public virtual void Clear();** Removes all elements from the Hashtable.
3	**public virtual bool ContainsKey(object key);** Determines whether the Hashtable contains a specific key.
4	**public virtual bool ContainsValue(object value);** Determines whether the Hashtable contains a specific value.
5	**public virtual void Remove(object key);** Removes the element with the specified key

Hashtable Class Example


```
using System;
using System.Collections;

namespace Chapter4
{
    class Program
    {
        static void Main(string[] args)
        {
            Hashtable ht = new Hashtable();
            ht.Add("001", "Zara Ali");
            ht.Add("002", "Abida Rehman");
            ht.Add("003", "Joe Holzner");
            ht.Add("004", "Mausam Benazir Nur");
            ht.Add("005", "M. Amlan");
            ht.Add("006", "M. Arif");
            ht.Add("007", "Ritesh Saikia");
            if (ht.ContainsValue("Nuha Ali"))
            {
                Console.WriteLine("This student name is already in the list");
            }
            else
            {
                ht.Add("008", "Nuha Ali");
            }
            // Get a collection of the keys.
            ICollection key = ht.Keys;
            foreach (string k in key)
            {
                Console.WriteLine(k + ": " + ht[k]);
            }
            Console.ReadKey();
        }
    }
}
```

OUTPUT

```
007: Ritesh Saikia
006: M. Arif
008: Nuha Ali
002: Abida Rehman
003: Joe Holzner
001: Zara Ali
004: Mausam Benazir Nur
005: M. Amlan
```

Hashtable Class Example

*for Loop


```
using System;
using System.Collections;

namespace Chapter4
{
    class Program
    {
        static void Main(string[] args)
        {
            Hashtable ht = new Hashtable();
            ht.Add("001", "Zara Ali");
            ht.Add("002", "Abida Rehman");
            ht.Add("003", "Joe Holzner");
            ht.Add("004", "Mausam Benazir Nur");
            ht.Add("005", "M. Amlan");
            ht.Add("006", "M. Arif");
            ht.Add("007", "Ritesh Saikia");
            if (ht.ContainsValue("Nuha Ali"))
            {
                Console.WriteLine("This student name is already in the list");
            }
            else
            {
                ht.Add("008", "Nuha Ali");
            }
            // Get a collection of the keys.

            ICollection keyCollection = ht.Keys;
            string[] keys = new string[keyCollection.Count];
            keyCollection.CopyTo(keys, 0);
            for (int i = 0; i < keys.Length; i++)
            {
                Console.WriteLine(keys[i] + ": " + ht[keys[i]]);

            }
            Console.ReadKey();
        }
    }
}
```

OUTPUT

```
007: Ritesh Saikia
006: M. Arif
008: Nuha Ali
002: Abida Rehman
003: Joe Holzner
001: Zara Ali
004: Mausam Benazir Nur
005: M. Amlan
```

Hashtable Class Example

*get a Data from a Key

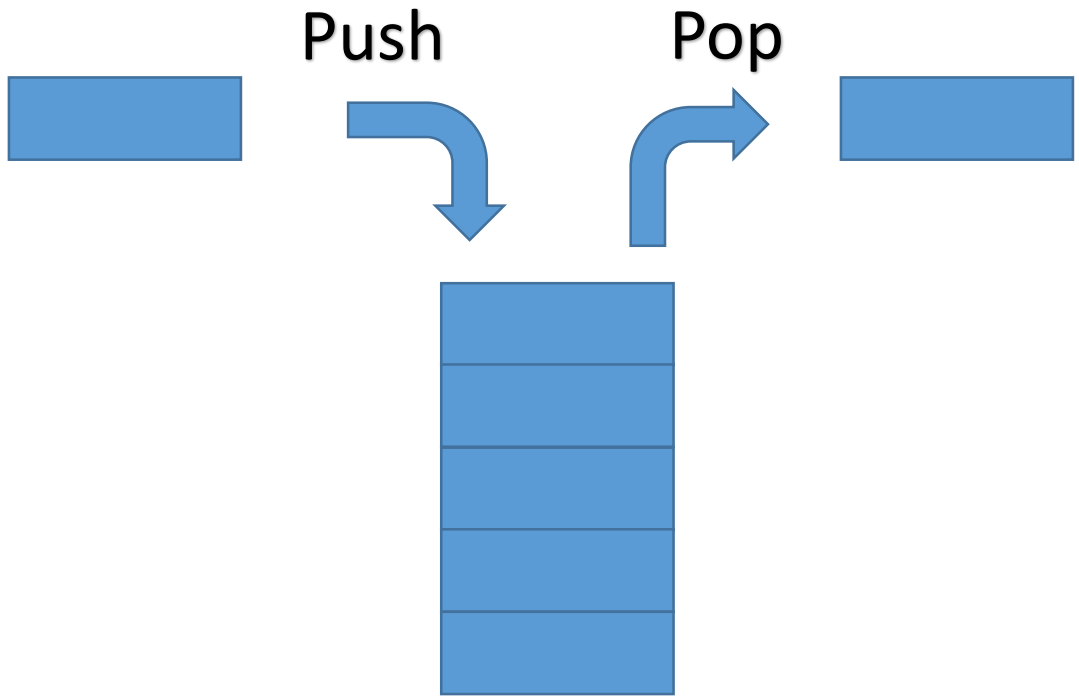

```
using System;
using System.Collections;

namespace Chapter4
{
    class Program
    {
        static void Main(string[] args)
        {
            Hashtable ht = new Hashtable();
            ht.Add("001", "Zara Ali");
            ht.Add("002", "Abida Rehman");
            ht.Add("003", "Joe Holzner");
            ht.Add("004", "Mausam Benazir Nur");
            ht.Add("005", "M. Amlan");
            ht.Add("006", "M. Arif");
            ht.Add("007", "Ritesh Saikia");
            if (ht.ContainsValue("Nuha Ali"))
            {
                Console.WriteLine("This student name is already in the list");
            }
            else
            {
                ht.Add("008", "Nuha Ali");
            }
            // Get a data from key.
            string find = "001";
            if (ht.ContainsKey(find))
            {
                Console.WriteLine("key: " + find + "  data: " +ht[find]);
            }
            Console.ReadKey();
        }
    }
}
```

OUTPUT

```
key: 001  data: Zara Ali
```

Stack Class

- Stack เป็นโครงสร้างแบบเรียงลำดับก่อนหลัง แต่ข้อมูลที่เข้าสู่ Stack ทีหลังจะออกก่อน เรียกว่า LIFO (Last In First Out)



Properties of the Stack Class

Property	Description
Count	Gets the number of elements contained in the Stack.

Methods of the Stack Class

No	Methods
1	**public virtual void Clear();** Removes all elements from the Stack.
2	**public virtual bool Contains(object obj);** Determines whether an element is in the Stack.
3	**public virtual object Peek();** Returns the object at the top of the Stack without removing it.
4	**public virtual object Pop();** Removes and returns the object at the top of the Stack.
5	**public virtual void Push(object obj);** Inserts an object at the top of the Stack.
6	**public virtual object[] ToArray();** Copies the Stack to a new array.

```
using System;
using System.Collections;

namespace Chapter4
{
    class Program
    {
        static void Main(string[] args)
        {

            Stack st = new Stack();
            st.Push('A');
            st.Push('M');
            st.Push('G');
            st.Push('W');
            Console.WriteLine("Current stack: ");
            foreach (char c in st)
            {
                Console.Write(c + " ");
            }
            Console.WriteLine();
            st.Push('V');
            st.Push('H');
            Console.WriteLine("The next poppable value in stack: {0}", st.Peek());
            Console.WriteLine("Current stack: ");
            foreach (char c in st)
            {
                Console.Write(c + " ");
            }
            Console.WriteLine();
            Console.WriteLine("Removing values ");
            st.Pop();
            st.Pop();
            st.Pop();
            Console.WriteLine("Current stack: ");
            foreach (char c in st)
            {
                Console.Write(c + " ");
            }
            Console.ReadKey();
        }
    }
}
```

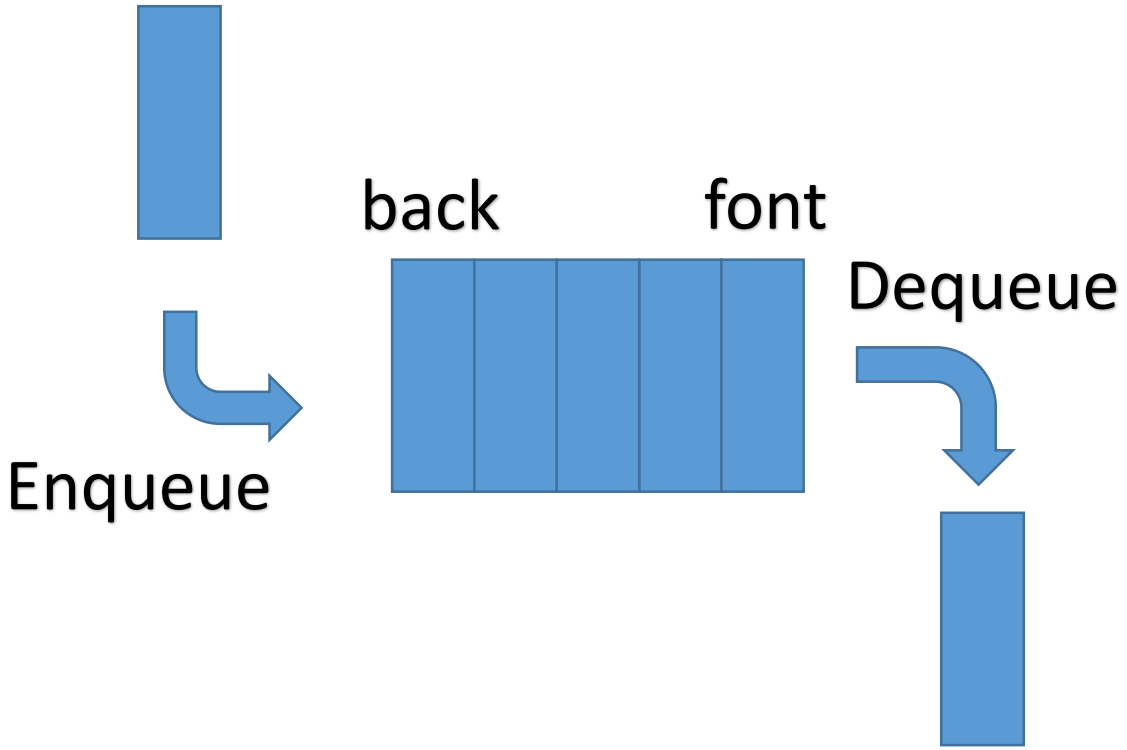
Stack Class Example

OUTPUT

```
Current stack:
W G M A
The next poppable value in stack: H
Current stack:
H V W G M A
Removing values
Current stack:
G M A
```

Queue Class

- Queue คือ Class ที่แทนโครงสร้างของข้อมูลที่มีการเรียงลำดับเช่นเดียวกันกับ Stack
- ข้อมูลที่เข้าสู้ Queue ก่อนจะออกจาก Queue ก่อนเสมอ เรียกว่า FIFO (First In First Out)



Properties of the Queue Class

Property	Description
Count	Gets the number of elements contained in the Queue.

Methods of the Queue Class

No	Methods
1	**public virtual void Clear();** Removes all elements from the Queue
2	**public virtual bool Contains(object obj);** Determines whether an element is in the Queue
3	**public virtual object Dequeue();** Removes and returns the object at the beginning of the Queue.
4	**public virtual void Enqueue(object obj);** Adds an object to the end of the Queue.
5	**public virtual object[] ToArray();** Copies the Queue to a new array.
6	**public virtual void TrimToSize();** Sets the capacity to the actual number of elements in the Queue.

```
using System;
using System.Collections;

namespace Chapter4
{
    class Program
    {
        static void Main(string[] args)
        {

            Queue q = new Queue();
            q.Enqueue('A');
            q.Enqueue('M');
            q.Enqueue('G');
            q.Enqueue('W');
            Console.WriteLine("Current queue: ");
            foreach (char c in q)
            {
                Console.Write(c + " ");
            }
            Console.WriteLine();
            q.Enqueue('V');
            q.Enqueue('H');
            Console.WriteLine("Current queue: ");
            foreach (char c in q)
            {
                Console.Write(c + " ");
            }
            Console.WriteLine();
            Console.WriteLine("Removing some values ");
            char ch = (char)q.Dequeue();
            Console.WriteLine("The removed value: " + ch);
            ch = (char)q.Dequeue();
            Console.WriteLine("The removed value: " + ch);
            Console.ReadKey();
        }
    }
}
```

Queue Class Example

OUTPUT

```
Current queue:
A M G W
Current queue:
A M G W V H
Removing values
The removed value: A
The removed value: M
```

แบบฝึกหัดที่ 1

- ให้นักศึกษาเขียนโปรแกรมแปลงจำนวนวันให้กลายเป็นเดือน

In Put Day : 89
89 days = 2 Month and 29 days

แบบฝึกหัดที่ 2

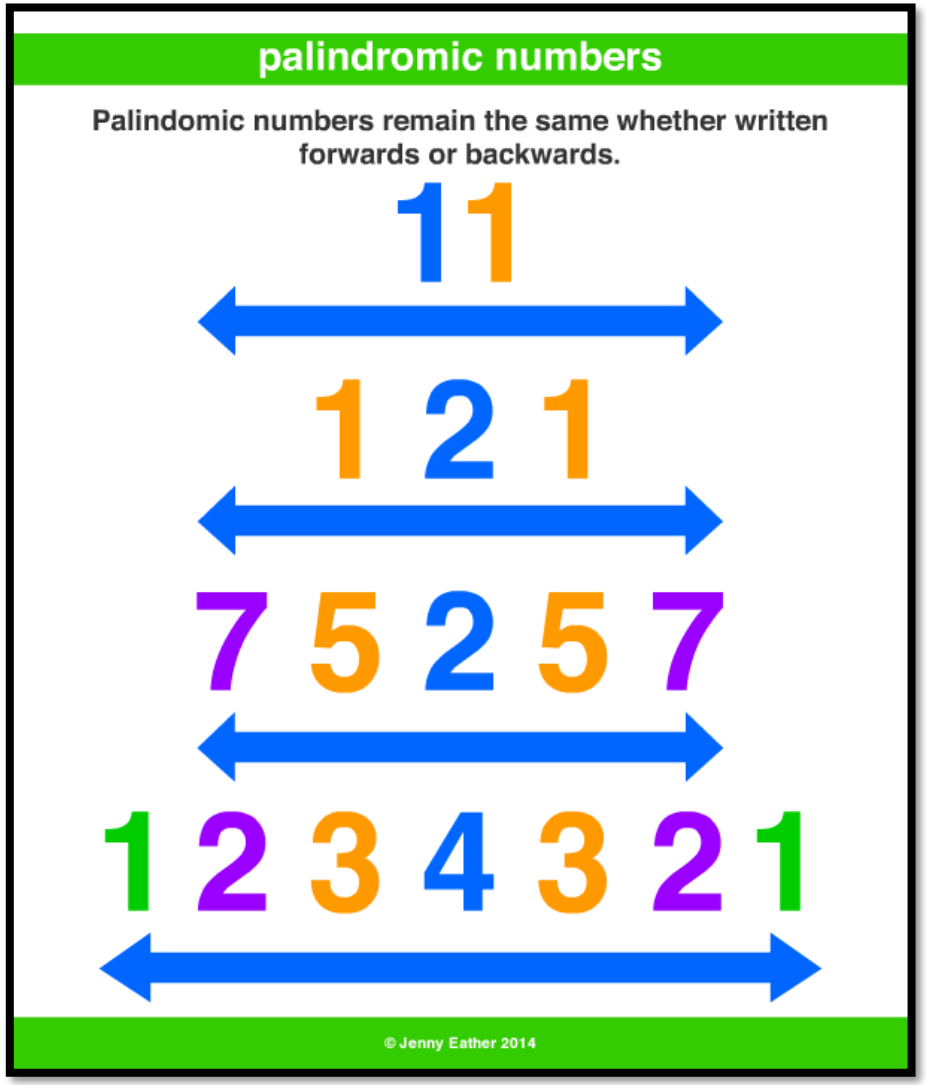
- จงเขียนโปรแกรมกลับด้านตัวเลข

input number : 123456
reverse is : 654321

แบบฝึกหัดที่ 3

- จงเขียนโปรแกรมตรวจสอบตัวเลข ว่าเป็น palindrome หรือไม่

input number : 12321
12321 is a Palindrome Number



CONCLUSION

การบ้านบทที่ 3

- จงเขียนโปรแกรมแสดงเลขคี่จากการวนซ้ำค่าตั้งแต่ 1 – 100 โดยใช้การวนซ้ำ
- จงยกตัวอย่างการใช้งาน Collections มาคนละ 1 ตัวอย่าง