

本周不辦理實作輔導
happy mother day!

Q1: 追蹤程式: 印出結果? 搶答

```
int p=0, i=1, n=9;
while (i<=n) {
    p=p*i;
    i=i+2;
}
System.out.println("p="+p+"n="+n);
```

Q2: 追蹤程式: 印出結果? 搶答

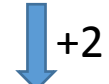
```
int p=0, i=1, q=9, s=0;  
for(p=100;p>50;p--) i++;  
for(;q<p;q++) s++;  
s=p+q;  
System.out.println("s="+s);
```

再談字元金字塔： nested loop 應用

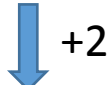
主題：字元金字塔 - 正金字塔

- 問題分析:

7個空白，印1個*



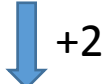
6個空白，印3個*



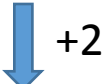
5個空白，印5個*



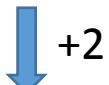
4個空白，印7個*



3個空白，印9個*



2個空白，印11個*

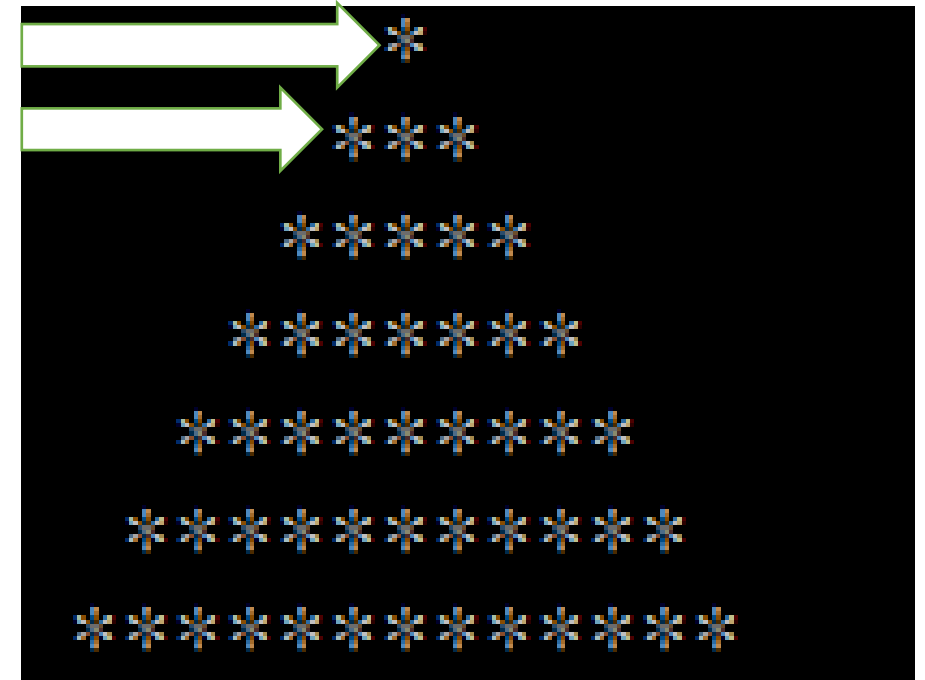


1個空白，印13個*

執行結果

7個空白

6個空白



主題：字元金字塔 – 正金字塔

- 利用迴圈印出「*」，逐行增加印出個數，直到印出7層金字塔。
- 本題承接上次巢狀(nested)迴圈的概念外，尚還利用到2個迴圈概念
- 由於金字塔每行更為增加2顆「*」，還有每行開頭空白數減少1的問題。因此在迴圈外都要考慮兩個更改的問題。

```
public class Charstar4 {  
    public static void main(String[] args) {  
        //變數level為金字塔的層數,每行*個數,每行空格數  
        int level = 7,star = 1,space = 7;  
  
        //for迴圈 直到印出 level行結束金字塔  
        for (int i = 0; i < level; i++) {  
            //印出空白  
            for (int j = 0; j < space; j++)  
                System.out.print(" ");  
            //在印出*  
            for (int j = 0; j < star; j++)  
                System.out.print("*");  
  
            star+=2;//每層星星數增二  
            space--;//每層結束空白數減一  
  
            //每層結束換行  
            System.out.println("");  
        }  
    }  
}
```

執行結果

```
 *  
***  
*****  
*****  
*****  
*****  
*****
```

主題：字元金字塔 - 正金字塔

```
public class Charstar4 {  
    public static void main(String[] args) {  
        //變數 level 為金字塔的層數, 每行 * 個數, 每行空格數  
        int level = 7, star = 1, space = 7;  
  
        //for 迴圈 直到印出 level 行結束金字塔  
        for (int i = 0; i < level; i++) {  
            //印出空白  
            for (int j = 0; j < space; j++)  
                System.out.print(" ");  
            //在印出 *  
            for (int j = 0; j < star; j++)  
                System.out.print("*");  
  
            star+=2; //每層星星數增二  
            space--; //每層結束空白數減一  
  
            //每層結束換行  
            System.out.println("");  
        }  
    }  
}
```

執行結果

7個空白

6個空白



字元金字塔 - 正金字塔II:減少變數使用 (找出空白、*個數與i層關係)

```
1. public class Charstar4_1b {  
2.     public static void main(String[] args) {  
3.         int level =7;  
4.         for (int i = 0; i < level; i++) {  
5.             for (int j = 0; j <= 7-i-1; j++)  
6.                 System.out.print(" ");  
7.             for (int k = 0; k <= 2*i; k++)  
8.                 System.out.print("*");  
9.             System.out.println("");  
10.        }  
11.  
12.    }  
13. }
```

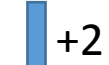
執行結果

```
      *  
     ***  
    *****  
   *********  
  ***********  
 *****  
*****
```

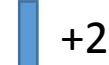

主題：字元金字塔 – 正金字塔 II(找出空白、*個數與i層關係)

- 問題分析:

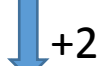
7個空白，印1個*： $i=0$ $j=0\sim 7-0-1$ $k=0\sim 2*0$



6個空白，印3個*： $i=1$ $j=0\sim 7-1-1$ $k=0\sim 2*1$



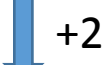
5個空白，印5個*： $i=2$ $j=0\sim 7-2-1$ $k=0\sim 2*2$



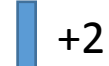
4個空白，印7個*： $i=3$ $j=0\sim 7-3-1$ $k=0\sim 2*3$



3個空白，印9個*： $i=4$ $j=0\sim 7-4-1$ $k=0\sim 2*4$



2個空白，印11個*： $i=5$ $j=0\sim 7-5-1$ $k=0\sim 2*5$

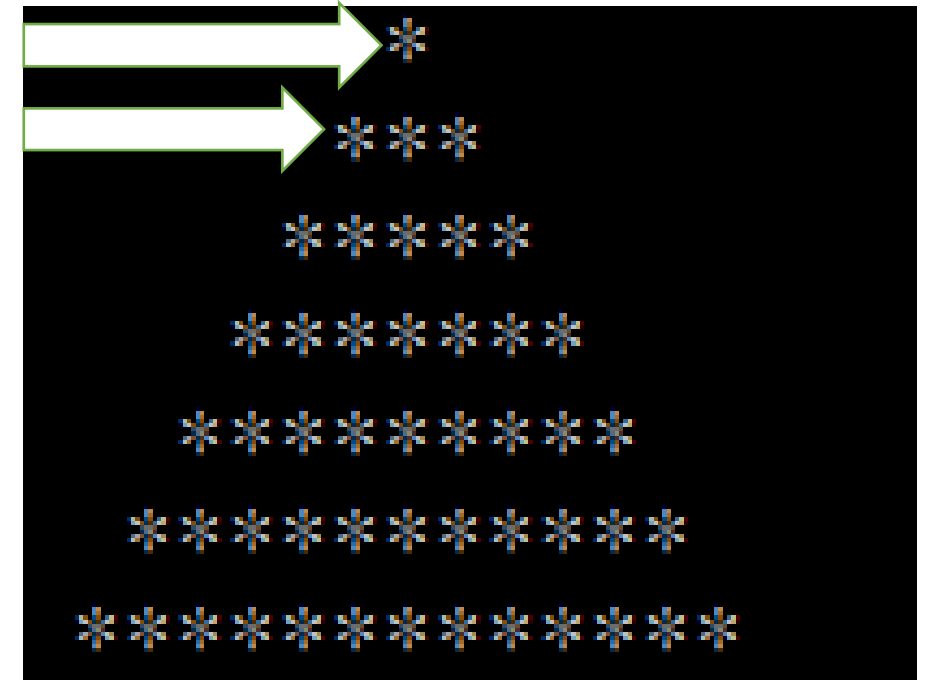


1個空白，印13個*： $i=6$ $j=0\sim 7-6-1$ $k=0\sim 2*6$

$$i=? \quad j=0\sim 7-i-1 \quad k=0\sim 2*i$$

執行結果

7個空白
6個空白



主題：字元金字塔

III- 使用者自訂“字元”及“層數”金字塔

執行結果

```
輸入字元：A
輸入層數：9
  A
 AAA
AAAAA
AAAAAAA
AAAAAAAAA
AAAAAAAAAAA
AAAAAAAAAAAAA
AAAAAAAAAAAAA
AAAAAAAAAAAAA
AAAAAAAAAAAAA
AAAAAAAAAAAAA
```

```
import java.util.Scanner;
public class Charstar5 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        //變數c為金字塔的字元
        char c;
        //變數level為金字塔的層數，變數space為前幾層的空格數，變數star為每層金字塔應數出的字元數
        int level, space, star= 1;

        System.out.print("輸入字元：");
        c = input.next().charAt(0);

        System.out.print("輸入層數：");
        level = input.nextInt();

        //初始space值等於level
        space = level;
        //for迴圈 直到印出level行結束金字塔
        for (int i = 0; i < level; i++) {
            for (int j = 0; j < space; j++)
                System.out.print(" ");
            for (int j = 0; j < star; j++)
                System.out.print(c);

            System.out.println("");
            star += 2;
            space--;
        }
    }
}
```

第12周習題:

- 主題:輸入n層(n須為奇數)及"字元"，印出n層三角形(如下)
 - 使用者自訂"字元"及"層數"
 - 須能重複執行，直至輸入0層
 - 繳交"設計歷程"檔及.java

輸入n層: 7
輸入字元: *

*

**

**

*

輸入n層: 5
輸入字元: #

#

##

###

##

#

第12周習題:

- 主題:輸入n層(n須為奇數)及"字元"，印出n層三角形(如下)
 - 使用者自訂"字元"及"層數"
 - 須能重複執行，直至輸入0層
 - 繳交"設計歷程"檔及.java

輸入n層: 7
輸入字元: *

```
*  
  
**  
  
***  
  
****  
  
***  
  
**  
  
*
```

輸入n層: 5
輸入字元: #

```
#  
  
##  
  
###  
  
##  
  
#
```

陣列(array)基本概念

臺北市立大學 [資訊科學系\(含碩士班\)](#)

賴阿福

利用大量變數，處理大量資料？

- 有何缺點及問題??

例子：輸入100筆整數資料，要求出平均數，如何做？若找最大值、要求變異數，如何做？

討論：

```
input.nextInt(x1);  
input.nextInt(x2);  
...  
input.nextInt(x100);  
avg=(x1+x2+...+x100)/100;  
Var= (x1-avg)* (x1-avg);  
Var=(x2-avg) (x2-avg);  
...  
Var=(x100-avg) (x100-avg);  
//最後變異數結果敘述
```

(1)需100行讀取資料之敘述

(2)須很長的累加運算敘述，需很長運算式解平均數，需100行找最大值、處理變異數之敘述。

(3)極為暴力方式(brute force)

(4)沒有彈性，例如要處理1000筆？Big Data？

問題剖析：資料記錄存放問題！

不可以只用大量變數而已。

體驗處理10筆資料

- 如果輸入100筆整數資料?
- code長度?
- 彈性?效率?
- 如何改善?

```
if (max<x2) max=x2;  
else max=max;
```

```
if (max<x2) max=x2; else max=max;  
max= (max<x2) ? x2:max;
```

```
import java.util.Scanner;  
public class Tendata_noarray{  
    public static void main(String[] args){  
        Scanner input = new Scanner(System.in);  
        int x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,mean,max;  
        x1=input.nextInt();  
        x2=input.nextInt();  
        x3=input.nextInt();  
        x4=input.nextInt();  
        x5=input.nextInt();  
        x6=input.nextInt();  
        x7=input.nextInt();  
        x8=input.nextInt();  
        x9=input.nextInt();  
        x10=input.nextInt();  
        mean=(x1+x2+x3+x4+x5+x6+x7+x8+x9+x10)/10;  
        max=x1;  
        max=(max<x2)?x2:max;  
        max=(max<x3)?x3:max;  
        max=(max<x4)?x4:max;  
        max=(max<x5)?x5:max;  
        max=(max<x6)?x6:max;  
        max=(max<x7)?x7:max;  
        max=(max<x8)?x8:max;  
        max=(max<x9)?x9:max;  
        max=(max<x10)?x10:max;  
  
        System.out.println("x1 = "+x1);  
        System.out.println("x2 = "+x2);  
        System.out.println("x3 = "+x3);  
        System.out.println("x4 = "+x4);  
        System.out.println("x5 = "+x5);  
        System.out.println("x6 = "+x6);  
        System.out.println("x7 = "+x7);  
        System.out.println("x8 = "+x8);  
        System.out.println("x9 = "+x9);  
        System.out.println("x10 = "+x10);  
        System.out.println("mean = "+mean);  
        System.out.println("max = "+max);  
    }  
}
```

```
E:\java-2017\11-3>java Tendata_noarray  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
x1 = 1  
x2 = 2  
x3 = 3  
x4 = 4  
x5 = 5  
x6 = 6  
x7 = 7  
x8 = 8  
x9 = 9  
x10 = 10  
mean = 5  
max = 10
```


輸入10筆整數資料(改用迴圈)，求出平均數及最大值(未用陣列)

```
import java.util.Scanner;
public class without_array {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int sc = 0, n=1, total=0, max=0;
        float mean;
        while (n<=10) {
            System.out.print("輸入第"+n+"筆分數：");
            sc=input.nextInt();
            total=total+sc;
            max=(max<sc)?sc:max;
            n++; }
        mean=total/10;
        System.out.println("最大值："+max);
        System.out.println("平均數："+mean);
    } //main
} //class
```

```
C:\Users\user\Desktop\11-3>java without_array
輸入第1筆分數：23
輸入第2筆分數：45
輸入第3筆分數：67
輸入第4筆分數：87
輸入第5筆分數：89
輸入第6筆分數：90
輸入第7筆分數：98
輸入第8筆分數：76
輸入第9筆分數：78
輸入第10筆分數：88
最大值：98
平均數：74.0
```

- 輸入**10**筆整數資料，求出平均數及最大值!
- 若要再求**變異數**，如何做？(impossible)
- 之後再求**min**，如何做？(impossible)
- 後續印出**10**筆整數資料，如何做？(String)

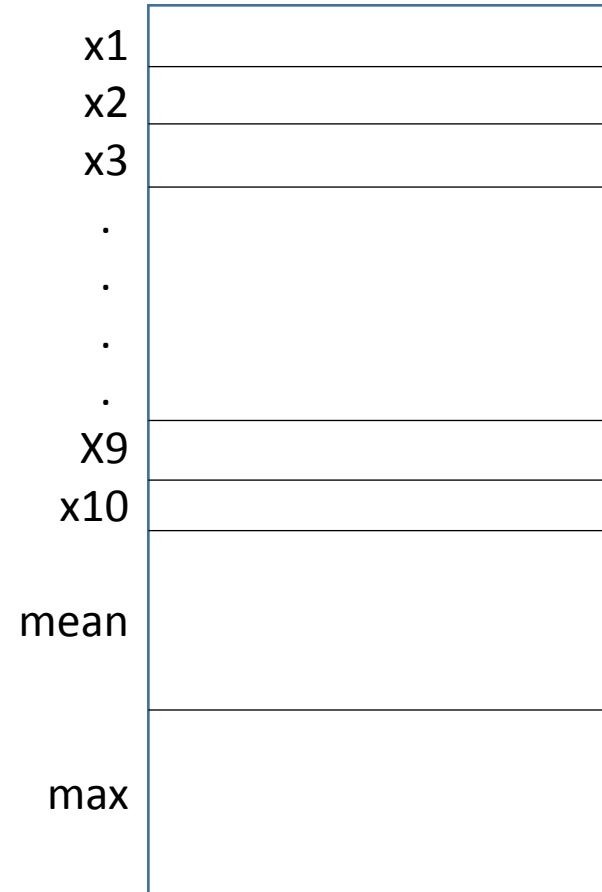
用字串存放輸入資料:字串連結

```
import java.util.Scanner;
public class without_array2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int sc = 0, n=1, total=0, max=0;
        float mean;
        String data="";
        while (n<=10) {
            System.out.print("輸入第"+n+"筆分數 : ");
            sc=input.nextInt();
            data=data+"第"+n+"筆分數 : "+sc+"\n";
            total=total+sc;
            max=(max<sc)?sc:max;
            n++; }
        mean=total/10;
        System.out.println("\n最大值: "+max);
        System.out.println("平均數: "+mean);
        System.out.println(data);
    } //main
} //class
```

```
C:\Users\user\Desktop\11-3>java without_array2
輸入第1筆分數 : 11
輸入第2筆分數 : 22
輸入第3筆分數 : 33
輸入第4筆分數 : 44
輸入第5筆分數 : 55
輸入第6筆分數 : 66
輸入第7筆分數 : 77
輸入第8筆分數 : 88
輸入第9筆分數 : 99
輸入第10筆分數 : 7
最大值: 99
平均數: 50.0
第1筆分數 : 11
第2筆分數 : 22
第3筆分數 : 33
第4筆分數 : 44
第5筆分數 : 55
第6筆分數 : 66
第7筆分數 : 77
第8筆分數 : 88
第9筆分數 : 99
```

利用多個變數，處理資料(未使用陣列)

- 任何時候，若再多一筆資料，需再增加程式碼，求 **max**、**mean** 敘述都須修改程式。
- 皆是序列(**Sequence**)，無法使用迴圈，缺乏自動化機制。
- 屬於暴力法。

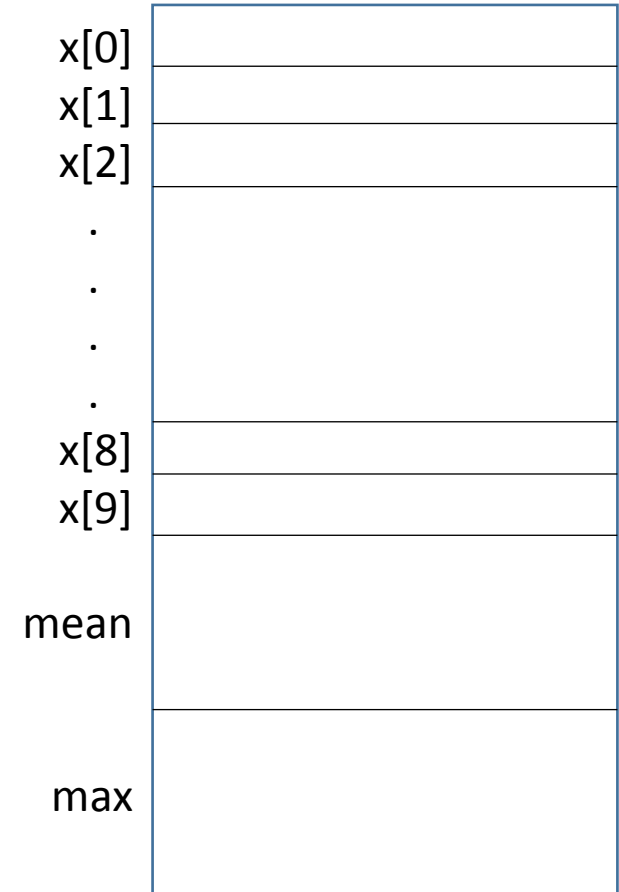


陣列(array):物以類聚

- 相同型態(ex. int, double...)的元素所形成有序的有限集合
- 陣列的元素被存放在**連續**的記憶體

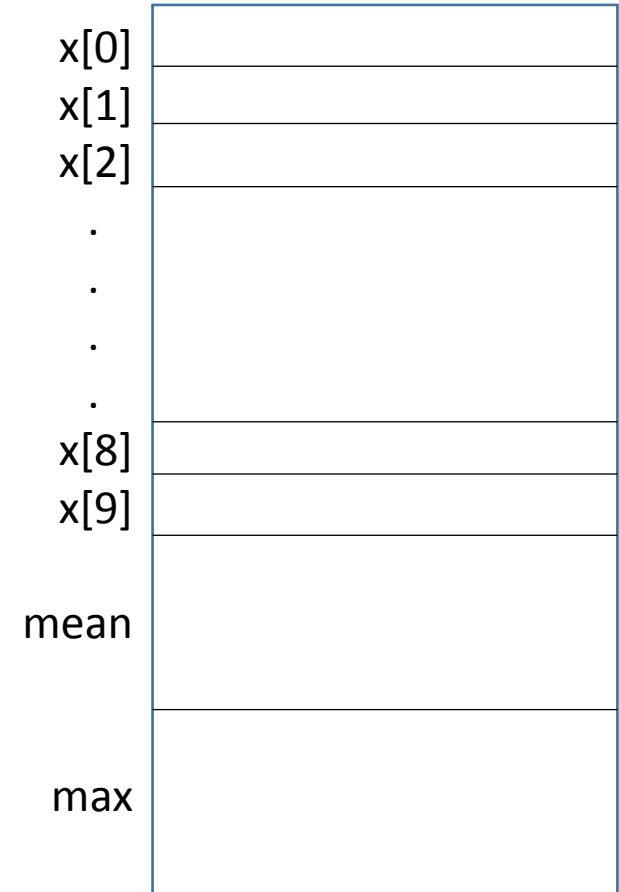
陣列

- 以一名稱代表一**序列資料的集合**，陣列**命名與變數命名相同**
- 以**索引(index)/註標(subscript)**來控制某一元素存取
 - 索引可用**變數**，因此可用**迴圈控制變數**，再控制陣列存取
 - **索引由0開始**。
- 宣告
`int [] x=new int[100];` //**索引:0~99**，**長度: 100**
- 陣列**長度**(陣列的屬性)；`x.length`
- 可在宣告時給予**初值**(不用指定長度)
`int [] sc={20,30,40};`
`String [] diages={"體重過輕Underweight","正常Normal","過重Overweight"};`

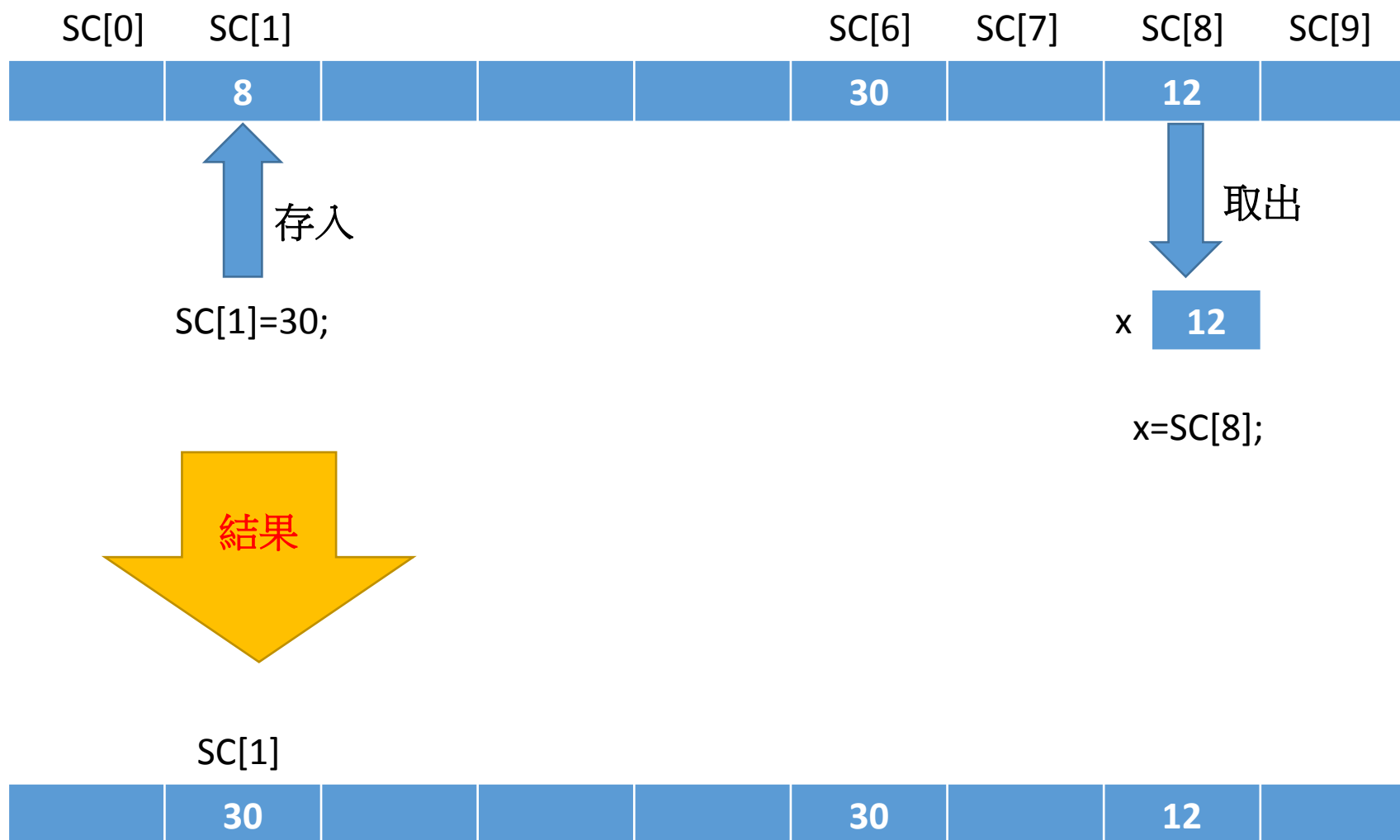


陣列的特性

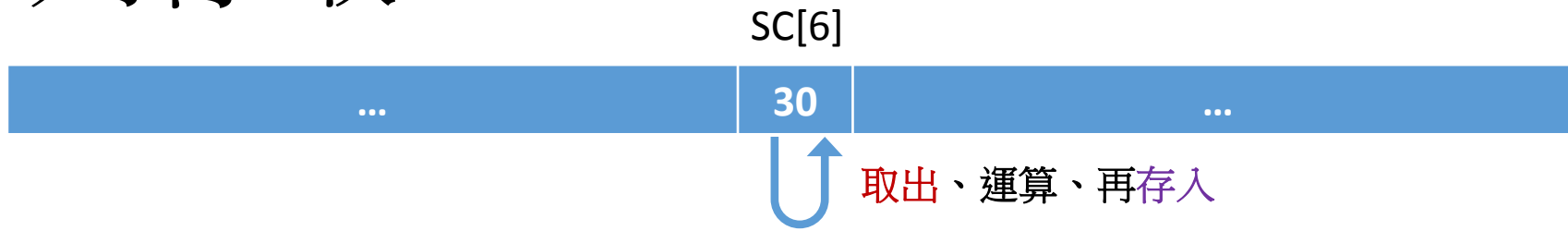
- 定義：相同型態(ex. int, double...)的元素所形成有序的有限集合
- 以索引值(index) 與 值(value) 來表示其對應關係。
- 通常陣列的元素被存放在**連續**的記憶體上，可以支援直接存取。
- 陣列元素存取時，將元素的索引/註標(Index/Subscript) 以位址函數(Address Function)計算出(對應)記憶體位址，再存取記憶體中的內容(元素的值)。
 - 以索引存取陣列元素之值(value)
 - `i=8;`
 - `x[i]=100;`
 - 存: `x[8]=100;` //將100存入x[8]
 - 取: `System.out.print (x[8]);` //取x[8]
 - `y=x[8]+x[9];` //取x[8]、x[9]
`x[7]=x[8]+x[9];`



陣列存取

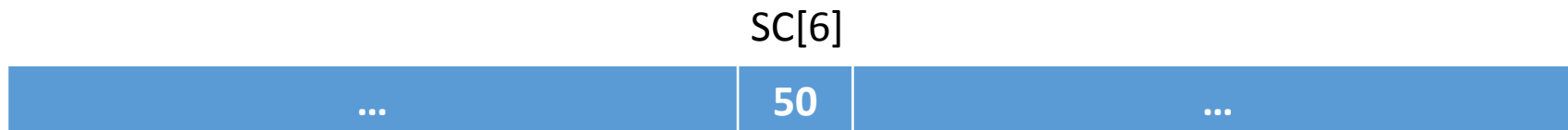


陣列存取



取出、運算、再存入

$SC[6] = SC[6] + 20;$



陣列應用:處理10筆資料

應用陣列存放**10筆資料**，then**處理**

```
1. public class Tendata_array0{
2.     public static void main(String[] args){
3.         int [] x={70,90,55,66,12,27,34,47,80,100};
4.         int mean=0,max=0,i=0, sum=0, flunk=0;
5.         for(i=0;i<x.length;i++) {
6.             System.out.println("x["+i+"] :"+x[i]);
7.             sum=sum+x[i];
8.             max=(max<x[i])?x[i]:max;
9.             if (x[i]<60) flunk++;//不及格人數
10.        }
11.        mean=sum/10;//平均數
12.        System.out.println("陣列長度: "+x.length);
13.        System.out.println("mean = "+mean);
14.        System.out.println("max = "+max);
15.        System.out.println("不及格人數: "+flunk);
16.    }//main
17. }//class
```

```
C:\Users\user\Desktop\11-3>java Tendata_array0
```

```
x[0] :70
x[1] :90
x[2] :55
x[3] :66
x[4] :12
x[5] :27
x[6] :34
x[7] :47
x[8] :80
x[9] :100
陣列長度: 10
mean = 58
max = 100
不及格人數: 5
```

x[0]	70
x[1]	90
x[2]	55
.	
.	
.	
.	
x[8]	80
x[9]	100
mean	
max	

未提供輸入資料

```
E:\java-2017\11-3>java Tendata_array1
```

```
input data for x[0] :80  
input data for x[1] :70  
input data for x[2] :80  
input data for x[3] :66  
input data for x[4] :77  
input data for x[5] :44  
input data for x[6] :55  
input data for x[7] :33  
input data for x[8] :22  
input data for x[9] :11
```

```
x[0] :80  
x[1] :70  
x[2] :80  
x[3] :66  
x[4] :77  
x[5] :44  
x[6] :55  
x[7] :33  
x[8] :22  
x[9] :11
```

```
mean = 53
```

```
max = 80
```

```
不及格人數: 5
```

```
import java.util.Scanner;  
public class Tendata_array1  
{  
    public static void main(String[] args){  
        Scanner input = new Scanner(System.in);  
        int [] x=new int[10];  
        int mean=0,max=0,i=0, sum=0, flunk=0;  
        while (i<=9) {  
            System.out.println("input data for x["+i+"] :");  
            x[i]=input.nextInt();  
            i++;}  
        System.out.print("\n");  
        for(i=0;i<x.length;i++) {  
            System.out.println("x["+i+"] :"+x[i]);  
            sum=sum+x[i];  
            max=(max<x[i])?x[i]:max;  
            if (x[i]<60) flunk++;  
        }  
        mean=sum/10;  
        System.out.println("mean = "+mean);  
        System.out.println("max = "+max);  
        System.out.println("不及格人數: "+flunk);  
    }  
}
```

```
import java.util.Scanner;
public class Tendata_array1{
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int [] x=new int[10];
        int mean=0,max=0,i=0, sum=0, flunk=0;
        while (i<=9) {
            System.out.print("input data for x["+i+"] :");
            x[i]=input.nextInt();
            i++;}
        System.out.print("\n");
        for(i=0;i<x.length;i++) {
            System.out.println("x["+i+"] :"+x[i]);
            sum=sum+x[i];
            max=(max<x[i])?x[i]:max;
            if (x[i]<60) flunk++;
            }
        mean=sum/10;
        System.out.println("mean = "+mean);
        System.out.println("max = "+max);
        System.out.println("不及格人數: "+flunk);
        //main
    }
//class
```

宣告陣列

輸入10筆資料:
i=0:第1筆放入x[0]
i=1:第2筆放入x[1]

i=9:第10筆放入x[9]

分析10筆資料:
印出10筆資料
加總
找最大值
算出不及格人數

運用陣列存放BMI狀態
運用陣列存放weekname

運用陣列存放BMI狀態

```
String [] diages={"體重過輕Underweight","正常Normal","過重Overweight"};
int status;

if (bmi < 18.5) status = 0;
else if (bmi < 24) status = 1;  //(bmi>=18.5 && bmi < 24)
else status = 2;

System.out.println("BMI : "+bmi+" , 狀態: "+diages[status]);
```

diages [0]	體重過輕Underweight
diages [1]	正常Normal
diages [2]	過重Overweight

RAM

```
import java.util.Scanner;

public class BMI_array {

public static void main(String[] args) {

String ok="Y";

System.out.println("=====歡迎量測體位=====");

Scanner input = new Scanner(System.in);

double height, weight;

String [] diages={"體重過輕Underweight","正常Normal","過重Overweight"};

int status;

while (ok.toUpperCase().equals("Y")) {

System.out.print("輸入身高 : ");

height = input.nextDouble();

System.out.print("輸入體重 : ");

weight = input.nextDouble();

double bmi = Math.round((weight/(height*height) )*100)/100.0;

if (bmi < 18.5) status = 0;

else if (bmi < 24) status = 1;  //(bmi>=18.5 && bmi < 24)

else status = 2;

System.out.println("BMI : "+bmi+" , 狀態: "+diages[status]);

System.out.print("(繼續(Y/N) : ");

ok= input.next().toUpperCase();

}

}

System.out.println("=====bye bye=====");

}

}

}
```

```

import java.util.Scanner;
public class weekname_2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String [] weekname= {"Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"};
        int week = 0;
        while (week>=0) {
            System.out.print("輸入星期幾?");
            week = input.nextInt();
            if (week >= 0 && week <=7) {
                switch (week) {
                    case 0:
                    case 7:
                        System.out.println(weekname[0]);
                        break;
                    case 1:
                        System.out.println(weekname[1]);
                        break;
                    case 2:
                        System.out.println(weekname[2]);
                        break;
                    case 3:
                        System.out.println(weekname[3]);
                        break;
                    case 4:
                        System.out.println(weekname[4]);
                        break;
                    case 5:
                        System.out.println(weekname[5]);
                        break;
                    case 6:
                        System.out.println(weekname[6]);
                        break;
                } //switch
            } //if
            else
                System.out.print("無法判讀\n");
        } //while
    } //main
} //class

```

- 運用陣列存放 weekname，看步道好處？

weekname[0]	Sunday
weekname[1]	Monday
weekname[2]	Tuesday
weekname[6]	Sunday

```
import java.util.Scanner;
public class weekname_3 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String [] weekname= {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"};
        int week = 0;
        while (week>=0) {
            System.out.print("輸入星期幾?");
            week = input.nextInt();
            if (week >= 0 && week <=7) {
                System.out.println(weekname[week]);
            } //if
            else
                System.out.println("無法判讀\n");
        } //while
    } //main
} //class
```


平行陣列(Parallel Arrays)

```
import java.util.Scanner;

public class weekname_array {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int week = 0;
        String [] wname={"Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"};
        String [] cnumber={"日","一","二","三","四","五","六","七"};

        while (week<=7) {
            System.out.print("輸入星期幾?");
            week = input.nextInt();
            if (week >= 0 && week <=7)
                System.out.println("輸入星期"+cnumber[week]+" ,英文為"+wname[week]);
            else
                System.out.println("無法判讀\n");
        }//while
    }//main
}//class
```

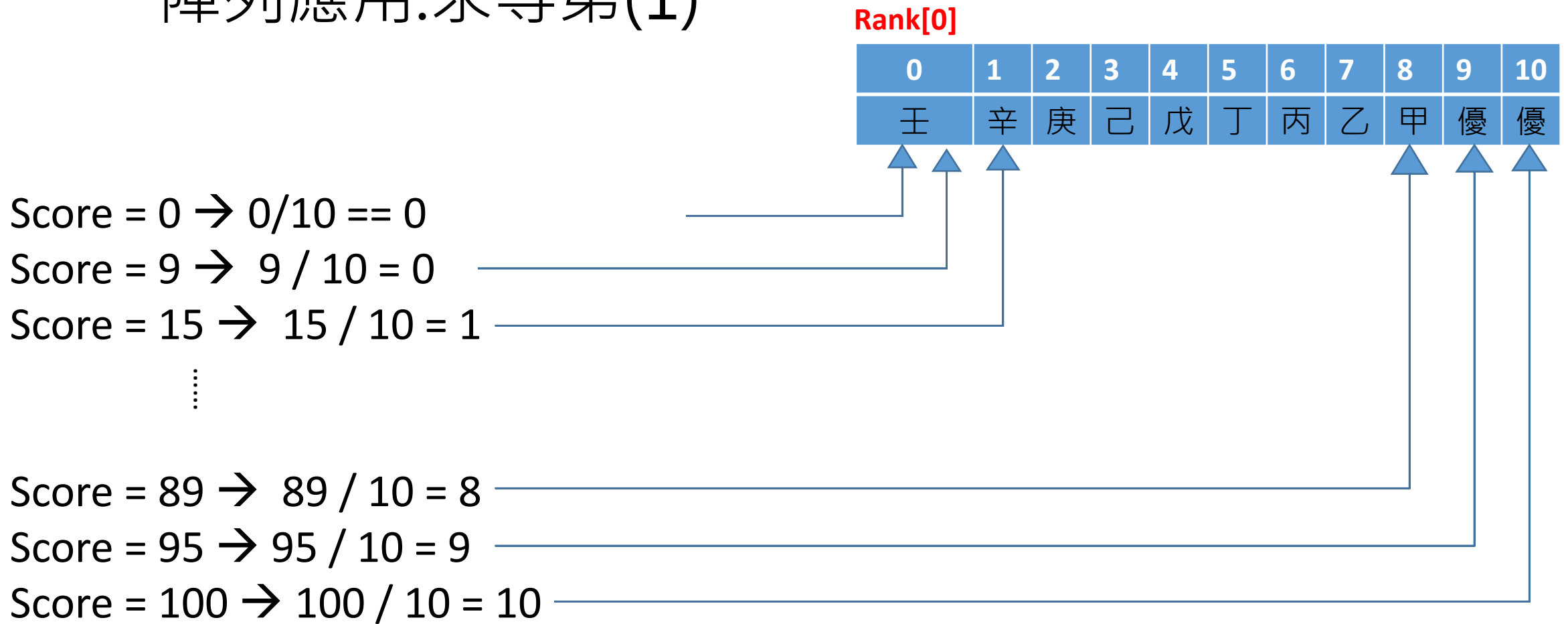
```
C:\Users\user\Desktop\11-3>java weekname_array
輸入星期幾?3
輸入星期三 ,英文為Wednesday
輸入星期幾?0
輸入星期日 ,英文為Sunday
輸入星期幾?7
輸入星期七 ,英文為Sunday
輸入星期幾?3
輸入星期三 ,英文為Wednesday
輸入星期幾?
```

陣列應用:求等第

陣列應用:求等第i

```
import java.util.Scanner;
public class scorerank_array {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int score = 0;
        String [] rank= {"王","辛","庚","己","戊","丁","丙","乙","甲","優","優"};
        while (score>=0) {
            System.out.print("輸入分數(整數, -1:end) : ");
            score = input.nextInt();
            //100~90優 89~80甲 79~70乙 69~60丙 59~50丁 49~40戊 39~30己 29~20庚 19~10辛 9~0王
            if (score >= 0)
                System.out.println("等第 : "+rank[score/10]);
            else
                System.out.println("無法判讀, bye!\n");
        }//while
    }//main
}//class
```

陣列應用:求等第(1)



*透過Score/10取得對應等第陣列索引，及Score/10計算結果為整數，作為索引(index)

*以空間換取時間，if之selection分支結構幾乎可忽略

陣列應用:求等第ii

```
import java.util.Scanner;
public class scorerank_array_2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int score = 0;
        String [] rank= {"優","優","甲","乙","丙","丁","戊","己","庚","辛","壬"};
        while (score>=0) {
            System.out.print("輸入分數(整數, -1:end) : ");
            score = input.nextInt();
            //100~90優 89~80甲 79~70乙 69~60丙 59~50丁 49~40戊 39~30己 29~20庚 19~10辛 9~0壬
            if (score<=100 && score >= 0)
                System.out.println("等第 : "+rank[10-(int)Math.floor(score/10)]);
            else if (score>100)
                System.out.println("無法判讀!\n");
            else
                System.out.println("bye!\n");
        }
    }
}
```

陣列應用:求等第(2)

0	1	2	3	4	5	6	7	8	9	10
優	優	甲	乙	丙	丁	戊	己	庚	辛	壬

Score = 100 $\rightarrow 10 - (100/10) == 0$

Score = 95 $\rightarrow 10 - (\text{int})\text{Math.floor}(95 / 10)$
= 1

Score = 89 $\rightarrow 10 - \text{Math.floor}(89 / 10) = 2$

Score = 9 $\rightarrow 10 - \text{Math.floor}(9 / 10) = 10$

Score = 0 $\rightarrow 10 - \text{Math.floor}(0/10) = 10$

*不同運算式，產生不同索引值，因此等第在陣列排列不同

* $(\text{int})\text{Math.floor}(95 / 10)$

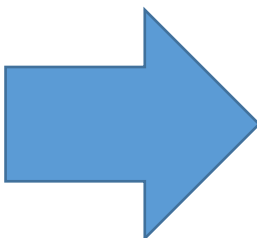
Math.floor(95 / 10) : double type

(int): type casting, convert double to int

```

switch (score / 10) {
    //由於100~90都為(10~9)優等，因此兩case印出相同
    case 10:
    case 9:
        System.out.print("等第：優\n");
        break;//結束執行,break switch判斷;
    case 8:
        System.out.print("等第：甲\n");
        break;
    case 7:
        System.out.print("等第：乙\n");
        break;
    case 6:
        System.out.print("等第：丙\n");
        break;
    case 5:
        System.out.print("等第：丁\n");
        break;
    case 4:
        System.out.print("等第：戊\n");
        break;
    case 3:
        System.out.print("等第：己\n");
        break;
    case 2:
        System.out.print("等第：庚\n");
        break;
    case 1:
        System.out.print("等第：辛\n");
        break;
    case 0://由於9~0除10結果均為0，但9~1 & 0屬不同等地，因此判斷後才印出
        if (score != 0)
            System.out.print("等第：壬\n");
        else
            System.out.print("等第：癸\n");
}

```



```

int score = 0;
String [] rank= {"優","優","甲","乙","丙","丁","戊","己","庚","辛","壬"};
while (score>=0) {
    System.out.print("輸入分數(整數, -1:end) : ");
    score = input.nextInt();
    //100~90優 89~80甲 79~70乙 69~60丙 59~50丁 49~40戊 39~30己 29~20庚
    if (score<=100 && score >= 0)
        System.out.println("等第： "+rank[10-(int)Math.floor(score/10)]);
    else if (score>100)
        System.out.println("無法判讀!\n");
    else
        System.out.println("bye!\n");
} //while

```

第周習題 (任選一題，亦可全做):將陣列加入第四周程式

習題A:輸入個人淨所得，求其應繳稅額 (陣列存放"稅率")

說明: (1)先上網搜尋個人淨所得之稅率資訊(須呈現於設計歷程檔中); (2)請繳交.java及設計歷程檔(.DOCX)。

習題B:輸入個人生日，求其星座及個性(陣列存放"星座資訊")

說明: (1)先上網搜尋12星座資訊(須呈現於設計歷程檔中);

(2)例如: **假設**處女座日期:8月23日~9月22日

- 輸入生日月份:9
- 輸入生日日期:1
- 結果:你的生日是**9月1日**，屬於**處女座**，個性:為人仔細，作事認真，對於是非善惡，判斷分明。

習題C:將BMI診斷分成六層次(陣列存放"六診斷層次")

例子：輸入100筆整數資料，要求出平均數，如何做？若要求變異數，如何做？ (c)

```
scanf("%d", &x1);  
scanf("%d", &x2);  
...  
scanf("%d", &x100);  
avg=(x1+x2+...+x100)/100;  
Var+=(x1-avg)^2;  
Var+=(x2-avg)^2;  
...  
Var+=(x100-avg)^2;  
//最後變異數結果敘述
```

```
int x[100], i,  
avg=0;  
for(i=0;i<100;i  
++)  
    scanf("%d",  
x[i]);  
for(i=0;i<100;i  
++)  
    avg=avg+x[i];  
avg=avg/100;
```

Review

switch case

改為5等第

```
import java.util.Scanner;
public class scorerank_2a {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int score = 0;
        while (score >= 0) {
            System.out.print("輸入分數(整數>=0) : ");
            score = input.nextInt();
            if (score < 0) break;
            //100~90優 89~80甲 79~70乙 69~60丙 59~0丁
            switch (score / 10) {
                case 10:
                case 9:
                    System.out.print("等第 : 優\n");
                    break; //結束執行, break switch判斷;
                case 8:
                    System.out.print("等第 : 甲\n");
                    break;
                case 7:
                    System.out.print("等第 : 乙\n");
                    break;
                case 6:
                    System.out.print("等第 : 丙\n");
                    break;
                case 5:
                case 4:
                case 3:
                case 2:
                case 1:
                case 0:
                    System.out.print("等第 : 丁\n");
                    break;
                default:
                    System.out.print("無法判讀\n");
                    break;
            } //switch
        } //while
    } //main
} //class
```

改變運算式

- $(score-50)/10$

```
import java.util.Scanner;
public class scorerank_2b {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int score = 0;
        while (score >= 0) {
            System.out.print("輸入分數(整數>=0) : ");
            score = input.nextInt();
            if (score < 0)
                {System.out.print("Bye\n");break;}
            //100~90優 89~80甲 79~70乙 69~60丙 59~0丁
            if (score <= 100 && score >= 0) {
                switch ((score-50)/10) {
                    case 5:
                    case 4:
                        System.out.print("等第 : 優\n");
                        break; //結束執行, break switch 判斷;
                    case 3:
                        System.out.print("等第 : 甲\n");
                        break;
                    case 2:
                        System.out.print("等第 : 乙\n");
                        break;
                    case 1:
                        System.out.print("等第 : 丙\n");
                        break;
                    case 0:
                    default:
                        System.out.print("等第 : 丁\n");
                        break;
                } //switch
            }
            else
                System.out.print("超過範圍\n");
        } //while
    } //main
} //class
```

質數： 不用boolean時

- import java.util.Scanner;
- public class prime_0 {
- static Scanner input = new Scanner(System.in);
- public static void main(String[] args) {
- System.out.println("====輸入>=2整數，判斷是否為質數?====");
- int n=3, i;
- String dif;
- //boolean prime;
- int prime;
- System.out.print("輸入>=2整數：");
- n = input.nextInt();
- i=2;
- //prime=true;
- prime=0;
- while (i<=n-1) {
- if (n%i==0)
- {prime=1; //prime=false;
- System.out.println(n+"可被"+i+"整除。");
- break;}
- else
- System.out.println(n+"不可被"+i+"整除。");
- ++i; }
- if (prime==0) dif="是質數!";
- else dif="不是質數!";
- System.out.println(n+dif);
- } //main
- } //class

Debug :輸入奇數n,求 $S=1+3+5+\dots+n$

```
1.import java.util.Scanner;
2.public class loop_debug_1 {
3. public static void main(String[] args) {
4. Scanner input = new Scanner(System.in);
5. int n=7,i=0, s=0;
6. System.out.println("輸入奇數n,求S=1+3+5+.....+n\n");
7. while (n>=1) {
8. System.out.print("輸入奇數(-1:end) : ");
9. n = input.nextInt();
10. if (n%2==0) {
11. System.out.println("輸入錯誤，須為奇數!");
12. continue;}
13. for(i=1;i<=n;++i)
14. s=s+i;
15. System.out.println("1+3+5+..."+n+"="+s);
16. }//while
17.
18. }//main
19.}//class
```

處理輸入錯誤