

Ex1

Given the code fragment:

```
String shirts[][] = new String[2][2];  
shirts[0][0] = "red";  
shirts[0][1] = "blue";  
shirts[1][0] = "small";  
shirts[1][1] = "medium";
```

Which code fragment prints red:blue:small:medium:?

A)

```
for (String[] c : shirts) {  
    for (String s : c) {  
        System.out.print(s + ":");  
    }  
}
```

B)

```
for (int index = 1; index < 2; index++) {  
    for (int idx = 1; idx < 2; idx++) {  
        System.out.print(shirts[index][idx] + ":");  
    }  
}
```

C)

```
for (int index = 0; index < 2; ++index) {  
    for (int idx = 0; idx < index; ++idx) {  
        System.out.print(shirts[index][idx] + ":");  
    }  
}
```

D)

```
for (int index = 0; index <= 2;) {  
    for (int idx = 0; idx <= 2;) {  
        System.out.print(shirts[index][idx] + ":");  
    }  
    index++;  
}
```

Ans:A

Ex2

Given:

```
class FieldInit {  
    Character c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println("c = " + c);  
        System.out.println("b = " + b);  
        System.out.println("f = " + f);  
    }  
}  
  
class test{  
    public static void main(String[] args) {  
        FieldInit f = new FieldInit();  
        f.printAll();  
    }  
}
```

What is the result?

A)

c = 0

b = false

f = 0.0F

B)

c =

b = false

f = 0.0

C)

c = null

b = true

f = 0.0

D)

c = null

b = false

f = 0.0

ans:D

Ex3

Given:

```
String stuff = "TV";
String res = null;
if (stuff.equals("TV")) {
    res = "Walter";
} else if (stuff.equals("Movie")) {
    res = "White";
} else {
    res = "No Result";
}
```

Which code fragment can replace the if block?

- A) `stuff.equals("TV") ? res = "Walter" : stuff.equals("Movie") ? res = "White " : res = "No Result";`
- B) `res = stuff.equals("TV") ? stuff.equals("Movie") ? "Walter" : "White" : "No Result";`
- C) `res = stuff.equals("TV") ? "Walter" else stuff.equals("Movie") ? "White ":"No Result";`
- D) `res = stuff.equals("TV") ? "Walter" : stuff.equals("Movie") ? "White" : "No Result";`

Ans:D

Ex4

Given the code fragment:

```
int array1[] = {1, 2, 3};
int array2[] = new int[5];
array2 = array1;
for (int i : array2) {
    System.out.print(i + " ");
}
System.out.println();
int array3[] = new int[3];
array3 = array2;
for (int i : array3) {
    System.out.print(i + " ");
}
```

What is the result?

- A)

1 2 3 0 0
1 2 3 0 0

B)

1 2 3 0 0
1 2 3

C)

1 2 3
1 2 3

D)

An Exception is thrown at run time.

Ans:C

Ex5

Given this array:

```
int[] intArr = {8, 16, 32, 64, 128};
```

Which two code fragments, independently, print each element in this array?

A)

```
for (int i=0 : intArr) {
    System.out.print(intArr[i] + " ");
    i++;
}
```

B)

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```

C)

```
for (int i : intArr) {
    System.out.print(intArr[i] + " ");
}
```

D)

```
for (int i : intArr) {
```

```
        System.out.print(i + " ");  
    }  
}
```

E)

```
for (int i=0; i < intArr.length; i++) {  
    System.out.print(i + " ");  
}
```

F)

```
for (int i; i < intArr.length; i++) {  
    System.out.print(intArr[i] + " ");  
}
```

Ans:BD

問題分析

這個問題是一個很經典的迴圈應用題目。題目要求，執行時使用者輸入列數，輸出結果會按照順序，1，3，5，7...且第一個為置中呈金字塔型，不是靠左對齊。

我們可以發現:第一行的左邊空格為 6 個，第二行為 5 個，第三行為 4 個，第四行為 3 個.....，每行依次遞減，直到最後一行空格數為 0;而星號數為第一行 1 個，第二行 3 個，第三行 5

個，.....，每行一次遞增 2，直到最後一行星號數為 13



撰寫步驟

Step 1:

依照問題分析得到，主要計算與最左差幾個空格的規律。得出結果如 5 列，第一行左邊就四個空格，剛好差一。

```
for(int j=1;j<=n-i;j++)//根據外層行號,輸出星號左邊空格數
{
    System.out.print(" ");
}
```

Step 2:

依照問題分析得到撰寫星號數，計算規則為當列 2 倍-1

```
for(int k=1;k<=2*i-1;k++)//根據外層行號,輸出星號個數
{
    System.out.print("*");
}
```

Step 3

完整程式

```
class ex3
{
    public static void main(String[] args) {
        java.util.Scanner sc=new java.util.Scanner(System.in);
        System.out.println("請輸入金字塔層數:");

        int n=sc.nextInt();
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=n-i;j++)//根據外層行號,輸出星號左邊空格數
            {
                System.out.print(" ");
            }
            for(int k=1;k<=2*i-1;k++)//根據外層行號,輸出星號個數
            {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Step 4:

編譯，測試執行

```
C:\UI>javac -encoding utf-8 ex3.java
```

```
C:\UI>java ex3
請輸入金字塔層數:
7
```

```

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

巴斯卡三角

根據輸入行數，列印出巴斯卡三角形。

```
C:\UI>java ex4
請輸入行數:9
1
11
121
1331
14641
15101051
1615201561
172135352171
18285670562881
```

問題分析:

觀察巴斯卡三角型的圖案，它的規律性為:三角形的直邊與斜邊都為“1”，三角形裡面任意數字正好等於前一行正上方與左上角兩個數字和，且第幾行就有幾個數字。

撰寫步驟

Step 1:

由問題分析可知，用二維陣列計算 t 無存巴斯卡三角，巴斯卡三角直邊與斜邊都設為 1，在設定中間元素直為上方與左上方相加的公式。以此撰寫一個 static 的方法 method 回傳值為此二維陣列，private static int[][] getTriangle(int num)

```
//得到巴斯卡三角方法
private static int[][] getTriangle(int num)//num為列數
{
    int[][] ary=new int[num][num]; //用二維陣列儲存

    for(int i=0;i<ary.length;i++)//直邊,斜邊設定為1
    {
        ary[i][0]=1; //直邊項目
        ary[i][i]=1; //斜邊項目
    }

    for(int i=1;i<ary.length;i++)//外迴圈控制行數
    {
        for(int j=1;j<=i;j++)//內迴圈控制列
        {
            ary[i][j]=ary[i-1][j-1]+ary[i-1][j]; //目前位置的正上方與左上方的和
        }
    }
    return ary;
}
```

Step 2:

撰寫列印巴斯卡三角的方法 static method，透過第一個方法產生的巴斯卡三角，為一二維陣列物件，所以這個方法設計成引數為陣列物件，只要將產生的二維陣列注入後，內容為將二維陣列內容輸出列印即可

```
//列印巴斯卡三角的方法
private static void print(int[][] ary)//注入產生的巴斯卡三角二位陣列物件
{
    for(int i=0;i<ary.length;i++)//外迴圈控制行
    {
        for(int j=0;j<=i;j++)//內迴圈控制列
        {
            System.out.print(ary[i][j]); //輸出內容
        }
        System.out.println(); //換行
    }
}
```

Step 3

完整程式

```

import java.util.Scanner;
class ex4
{
    //得到巴斯卡三角方法
    private static int[][] getTriangle(int num)//num為列數
    {
        int[][] ary=new int[num][num]; //用二維陣列儲存

        for(int i=0;i<ary.length;i++)//直邊,斜邊設定為1
        {
            ary[i][0]=1; //直邊項目
            ary[i][i]=1; //斜邊項目
        }

        for(int i=1;i<ary.length;i++)//外迴圈控制行數
        {
            for(int j=1;j<=i;j++)//內迴圈控制列
            {
                ary[i][j]=ary[i-1][j-1]+ary[i-1][j]; //目前位置的正上方與左上方的和
            }
        }
        return ary;
    }
}

```

```

//列印巴斯卡三角的方法
private static void print(int[][] ary)//注入產生的巴斯卡三角二位陣列物件
{
    for(int i=0;i<ary.length;i++)//外迴圈控制行
    {
        for(int j=0;j<=i;j++)//內迴圈控制列
        {
            System.out.print(ary[i][j]); //輸出內容
        }
        System.out.println(); //換行
    }
}

public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    System.out.print("請輸入行數:"); //鍵盤接收行數
    int num=sc.nextInt();

    int[][] ary=getTriangle(num); //使用撰寫好的方法-->得到巴斯卡三角
    print(ary); //使用列印輸出巴斯卡三角的方法

}
}

```

Step 4

編譯，測試與執行

```
C:\UI>javac -encoding utf-8 ex4.java
```

```
C:\UI>java ex4
```

請輸入行數:7

1

11

121

1331

14641

15101051

1615201561