

- 本周末安排實作輔導
- 預定：下周六

迴圈LOOP應用

- 判斷質數(**Prime number**)
- 求兩個整數的最大公因數(**greatest common divisor, GCD**)
- 判斷迴文(**palindrome**)

搶答!!

Q1 : 印出結果?

```
int s,x;
```

```
    s=0;
```

```
    for(x=1;x<5;x++)
```

```
        s=s*x;
```

```
    System.out.println("s="+s);
```

Q 2 : 印出結果?

```
s=1;
```

```
for(x=1;x<5;x++){
```

```
    s=s*x;
```

```
    x++;
```

```
}
```

```
System.out.println("s="+s);
```

Q 3 : 印出結果?

```
s=x=1;
  for(;x<100;){
    s=s+x;
    if(x==51)break;
    ++x;
  }
System.out.println("x="+x);
```

Q 4 : 印出結果?

```
s=x=1;
```

```
do{
```

```
    s=s+x;
```

```
    x++;
```

```
}while(x>=10);
```

```
System.out.println("x="+x+" s=" +s);
```

Q 5 : 印出結果?

```
s=x=10;
```

```
while(x<=0){
```

```
    s=x;
```

```
    x=x+2; }
```

```
System.out.println("x="+x+" s="+s);
```


Q 6 : 印出結果?

```
s=x=10;
while(x<=10){
    s=x;
    x=x+2;
}
System.out.println("x="+x+" s=" +s);
}
```

Q 1 ~ Q 6

初值

```
s=0  
s=3  
x=51  
x=2 s=2  
x=10 s=10  
x=12 s=10
```

```
public class sloop_test_A{  
    public static void main(String[] args){  
        int s,x;  
        s=0;  
        for(x=1;x<5;x++){  
            s=s*x;  
            System.out.println("s="+s);  
        }  
        s=1;  
        for(x=1;x<5;x++){  
            s=s*x;  
            x++;  
        }  
        System.out.println("s="+s);  
  
        s=x=1;  
        for(;x<100;){  
            s=s+x;  
            if(x==51)break;  
            ++x;  
        }  
        System.out.println("x="+x);  
  
        s=x=1;  
        do{  
            s=s+x;  
            x++;  
        }while(x>=10);  
        System.out.println("x="+x+" s=" +s);  
  
        s=x=10;  
        while(x<=0){  
            s=x;  
            x=x+2;    }  
        System.out.println("x="+x+" s=" +s);  
  
        s=x=10;  
        while(x<=10){  
            s=x;  
            x=x+2;  
        }  
        System.out.println("x="+x+" s=" +s);  
    }  
}
```

Q 7 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. for(i=1;i<=n;++i)
11. s=s+i;
12. System.out.println("1+3+5+...+"+n+"="+s);
13. }//while
14.
15. }//main
16.}//class
```

搶答：
那些錯誤?如何修?
寫出編號及修改結果

Q 8 Debug :輸入整數n,求 $S=1*2*3*.....*n$

```
1. import java.util.Scanner;
2. public class all_loop_1 {
3.     public static void main(String[] args) {
4.         Scanner input = new Scanner(System.in);
5.         int n=7,i=0;
6.         int s=0;
7.         System.out.println("輸入整數n,求 $S=1*2*3*.....*n$ \n");
8.         while (n>=1) {
9.             System.out.print("輸入整數(-1:end) : ");
10.            n = input.nextInt();
11.            for(i=1;i<=n;++i)
12.                s=s*i;
13.            System.out.println("1*2*3*...*"+n+"="+s);
14.        }//while
15.    }//main
16. }//class
```

搶答：
那些錯誤?如何修?
寫出編號及修改結果

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

不同for loop類型

四類型都equivalence

類型1(最常見):

```
①;  
for(①; ②; ③)  
  ④;  
⑤;
```

類型3:

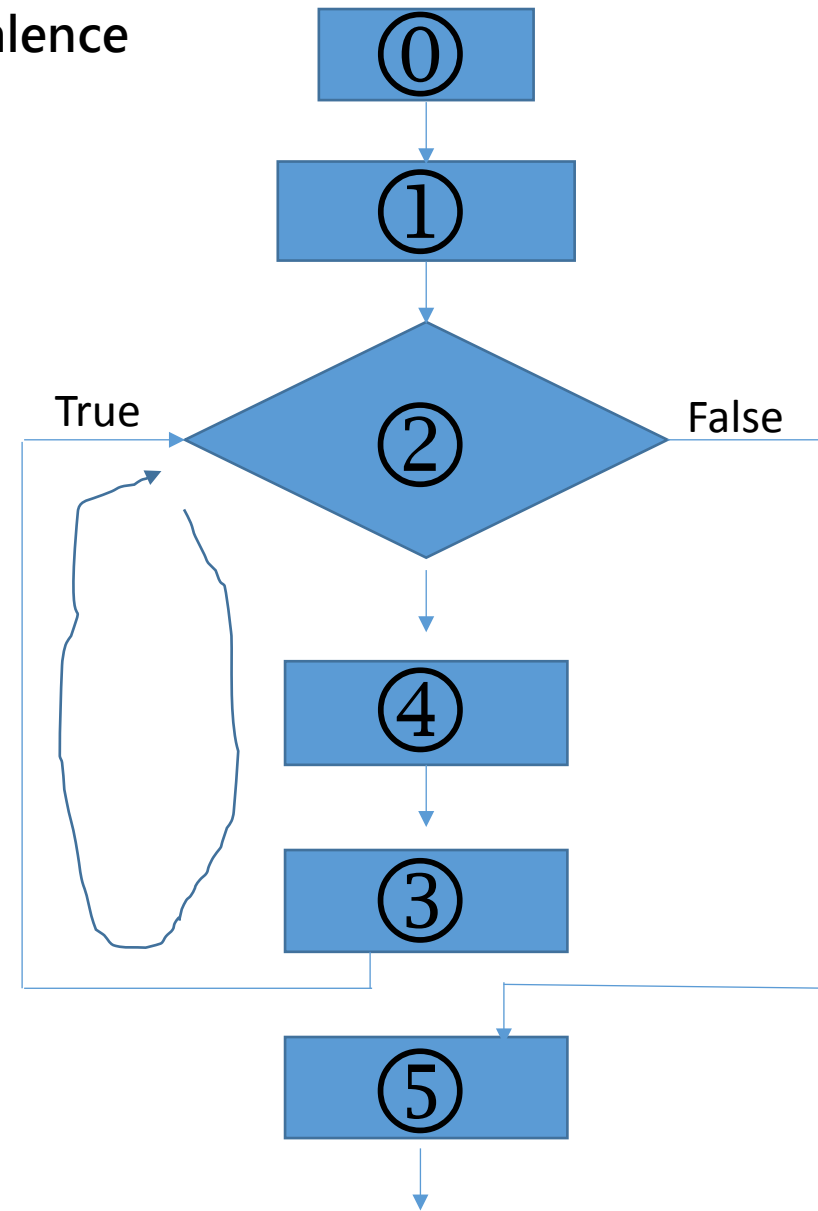
```
①;  
for(①; ②; ④, ③)  
;  
⑤;
```

類型2:

```
①;  
①;  
for(; ②; ③)  
  ④;  
⑤;
```

類型4:

```
①;  
①;  
for(; ②; )  
  {④; ③;}  
⑤;
```



迴圈LOOP應用

- 判斷質數
- 求兩個整數的最大公因數(greatest common divisor, GCD)
- 判斷迴文palindrome

判斷質數

• 質數(Prime number)

- 又稱**素數**，指在大於1的**自然數**中，除了1和該數自身外，無法被其他自然數**整除**的數（也可定義為只有1與該數本身兩個正**因數**的數）。
- 2, 3, 5, 7, 11, 13,是質數

• 解法:求n是否為質數?

• 利用除法

- $N/2, N/3, N/4, \dots, N/(N-1)$
 - $N \% 2 \neq 0, N \% 3 \neq 0, \dots, N \% (N-1) \neq 0$ (如何判斷整除)
- $N/2, N/3, N/4, \dots, N/(N/2)$

• 試除法

- 較簡單，但需時較長：
- 設被測試的自然數為n，使用此方法者需逐一測試2與 $\text{SQRT}(n)$ (\sqrt{n})之間的整數，確保它們無一能整除n。

判斷質數1

```
====輸入>=2整數，判斷是否為質數?====  
輸入>=2整數：23 輸入>=2整數：49  
23不可被2整除。 49不可被2整除。  
23不可被3整除。 49不可被3整除。  
23不可被4整除。 49不可被4整除。  
23不可被5整除。 49不可被5整除。  
23不可被6整除。 49不可被6整除。  
23不可被7整除。 49不可被7整除。  
23不可被8整除。 49不可被8整除。  
23不可被9整除。 49不可被9整除。  
23不可被10整除。 49不可被10整除。  
23不可被11整除。 49不可被11整除。  
23不可被12整除。 49不可被12整除。  
23不可被13整除。 49不可被13整除。  
23不可被14整除。 49不可被14整除。  
23不可被15整除。 49不可被15整除。  
23不可被16整除。 49不可被16整除。  
23不可被17整除。 49不可被17整除。  
23不可被18整除。 49不可被18整除。  
23不可被19整除。 49不可被19整除。  
23不可被20整除。 49不可被20整除。  
23不可被21整除。 49不可被21整除。  
23不可被22整除。 49不可被22整除。  
23是質數! 49不是質數!
```

```
int n=3, i;  
String dif;  
boolean prime;  
System.out.print("輸入>=2整數：");  
    n = input.nextInt();  
    if (n<=0) System.exit(-1);  
    i=2;  
    prime=true;  
    while (i<=n-1) {  
        if (n%i==0)  
            { prime=false;  
              break;}  
        ++i;  
    }  
    if (prime) dif="是質數!";  
    else dif="不是質數!";  
    System.out.println(n+dif);  
} //while
```

```
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;
        System.out.print("輸入>=2整數：");
        n = input.nextInt();
        i=2;
        prime=true;
        while (i<=n-1) {
            if (n%i==0)
                {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) dif="是質數!";
        else      dif="不是質數!";
        System.out.println(n+dif);
    } //main
} //class
```

判斷質數2:

- 減少除法次數
- 利用二層while，提供重複數入數值及判斷

```
int n=3, i;  
String dif;  
boolean prime;  
while (n>2) {  
    System.out.print("輸入>=2整數 : ");  
  
    n = input.nextInt();  
    if (n<=0) break;  
    i=2;  
    prime=true;  
    while (i*i<=n) {  
        if (n%i==0) {prime=false; break;}  
        ++i;  
    }  
    if (prime) dif="是質數!";  
    else dif="不是質數!";  
    System.out.println(n+dif);  
} //while
```

Demo判斷質數

- prime_0.java
- prime_1.java

求兩個整數的最大公因數(greatest common divisor, GCD)

- 兩個整數的最大公因數(greatest common divisor)是能夠同時整除它們的最大的正整數
- 求兩個整數GCD的方法:
 - 從2開始找，直到能整除兩個整數的最大正整數
 - 何時結束 (不會超過兩個整數的最小整數)
 - 200, 40的GCD
 - 輾轉相除法(下一回)
 - 輾轉相減法(下一回)
- 最小公倍數(LCM): $n1 * n2 / gcd$

求兩個整數GCD的方法1

- 兩個整數的最大公因數(**g**reatest **c**ommon **d**ivisor)是能夠同時整除它們的最大的正整數

99 and 363能被3整除,so 目前gcd改為3

99 and 363能被11整除,so 目前gcd改為11

99 and 363能被33整除,so 目前gcd改為33

1-while: $\text{GCD}(99, 363)=33$

1-while: $\text{LCM}(99, 363)=1089$

56 and 77能被7整除,so 目前gcd改為7

1-while: $\text{GCD}(56, 77)=7$

1-while: $\text{LCM}(56, 77)=616$

用三種loop完成: while

```
gcd=1; i=2;
```

```
while (i<=n1 && i<=n2) {
```

```
    if (n1%i==0 && n2%i==0)
```

```
        gcd=i;
```

```
    i++;} //while
```

```
System.out.println("1-while: GCD("+n1+", "+n2+")="+gcd);
```

```
System.out.println("1-while: LCM("+n1+", "+n2+")="+n1*n2/gcd);
```

用三種loop完成: do while

```
gcd=1; i=2;
```

```
do {
```

```
    if (n1%i==0 && n2%i==0)
```

```
        gcd=i;
```

```
    i++;
```

```
    } while (i<=n1 && i<=n2);
```

```
System.out.println("2-do while: GCD("+n1+", "+n2+")="+gcd);
```


用三種loop完成: for

```
for(gcd=1, i=2;(i<=n1 && i<=n2);i++)
```

```
    if (n1%i==0 && n2%i==0)
```

```
        gcd=i;
```

```
System.out.println("3-for: GCD("+n1+", "+n2+")="+gcd+"\n");
```

用三種while loop 顯示過程

```
gcd=1; i=2;
```

```
while (i<=n1 && i<=n2) {
```

```
    if (n1%i==0 && n2%i==0) {
```

```
        System.out.println(n1+" and "+n2+"能被"+i+"整除,so 目前gcd改為"+i);
```

```
        gcd=i;}
```

```
    i++;} //while
```

```
System.out.println("1-while: GCD("+n1+", "+n2+")="+gcd);
```

```
System.out.println("1-while: LCM("+n1+", "+n2+")="+n1*n2/gcd);
```

```

import java.util.Scanner;
public class gcd_1 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int n1=1, n2=1, gcd=1, i=2;
        System.out.println("輸入整數n1 & n2,求GCD(n1, n2) and LCM(n1, n2)\n");
        while (true) {
            gcd=1; i=2;
            System.out.print("輸入整數n1 : ");
            n1 = input.nextInt();
            System.out.print("輸入整數n2 : ");
            n2 = input.nextInt();
            if (n1<1 || n2<1) {
                System.out.println("bye!!");
                break;
            }
            while (i<=n1 && i<=n2) {
                if (n1%i==0 && n2%i==0) gcd=i;
                i++;} //while
            System.out.println("1-while: GCD("+n1+", "+n2+")="+gcd);
            System.out.println("1-while: LCM("+n1+", "+n2+")="+n1*n2/gcd);

            gcd=1; i=2;
            do {
                if (n1%i==0 && n2%i==0) gcd=i;
                i++;} while (i<=n1 && i<=n2);
            System.out.println("2-do while: GCD("+n1+", "+n2+")="+gcd);
            for(gcd=1, i=2;(i<=n1 && i<=n2);i++)
                if (n1%i==0 && n2%i==0) gcd=i;
            System.out.println("3-for: GCD("+n1+", "+n2+")="+gcd+"\n");

            //show the process of obtaining GCD
            gcd=1; i=2;
            while (i<=n1 && i<=n2) {
                if (n1%i==0 && n2%i==0) {
                    System.out.println(n1+" and "+n2+"能被"+i+"整除,so 目前gcd改為"+i);
                    gcd=i;}
                i++;} //while
            System.out.println("1-while: GCD("+n1+", "+n2+")="+gcd);
            System.out.println("1-while: LCM("+n1+", "+n2+")="+n1*n2/gcd);

        } //while
    } //main
} //class

```

palindrome 迴文

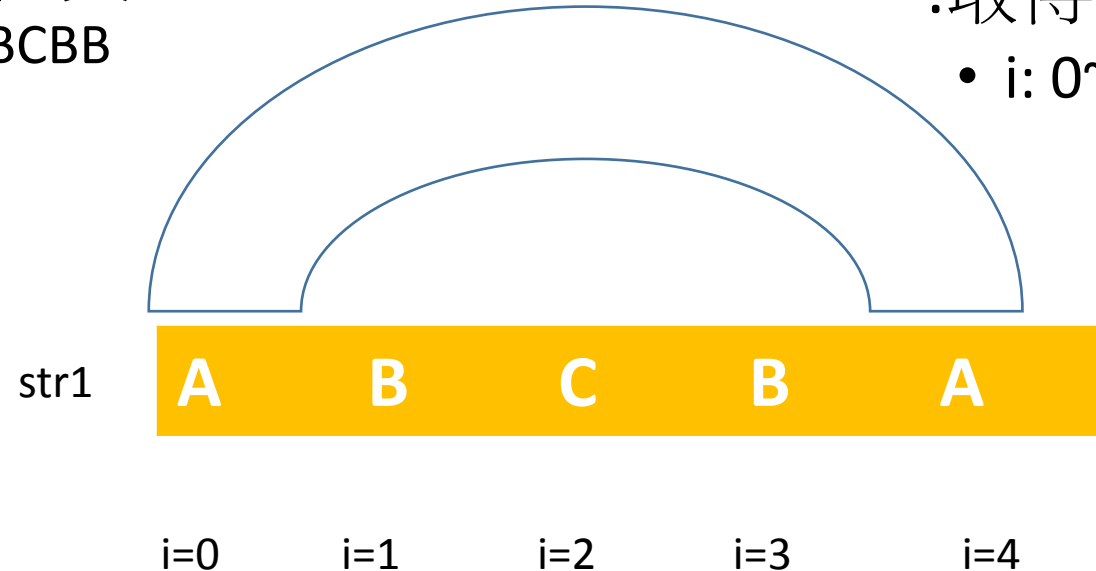
- 2017/1/2 line



palindrome迴文

- 輸入字串，判斷是否為迴文?
- 迴文
 - 2017102
 - ABCBA
 - ABBA
- 不是迴文
 - ABCBB

- 輸入字串
`Scanner input = new Scanner(System.in);`
`String str1 = input.nextLine();`
- `str1.length()` :字串長度
- `str1.charAt(i)` :取得字串第i個字元(`character`)
 - `i`: 0~ -1



`str1.length() == 5`

`str1.charAt(0)`

`str1.charAt(4)`

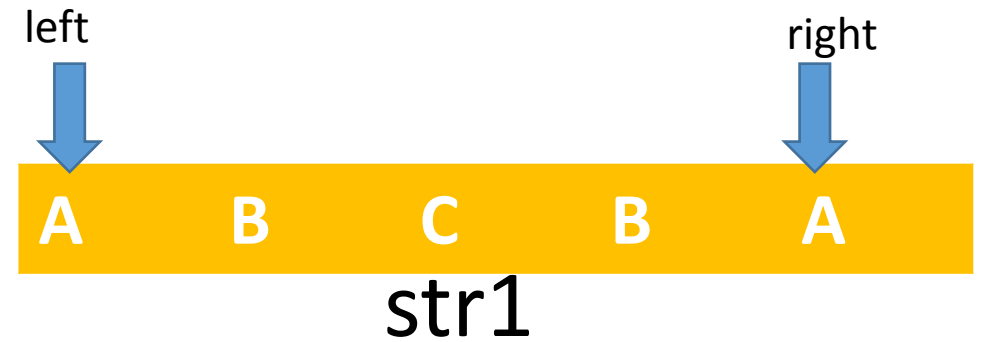
取得字串第i個字元(character)

```
System.out.print("輸入字串：");  
str1 = input.nextLine();  
if (str1.length()==0) break;  
int left=0, right=0, i=0;  
boolean palindrome=false;  
while (i<=str1.length()-1) {  
    System.out.print(str1.charAt(i));  
    i++;}
```

```
System.out.print("\n");  
i=str1.length()-1;  
while (i>=0) {  
    System.out.print(str1.charAt(i));  
    i--;}  
System.out.println("\n");
```

輸入字串，判斷是否為迴文？

```
left=0;right=str1.length()-1;
boolean palindrome=true;
while (left<right) {
    if (str1.charAt(left)!=str1.charAt(right))
        {palindrome=false;
         break;}
    left++; right--;
} //while
if (palindrome) dif="是迴文!";
else          dif="不是迴文!";
System.out.println(+dif);
```



第10周習題:

- **10 -1:求 $2 \sim n$ 之所有質數**
 - 輸入 n ，列出 $2 \sim n$ 之所有質數
 - 輸入錯誤(如 ≤ 1)，結束程式
 - 輸入 n 後，分別以 `while`, `do..while`, `for` 等三種迴圈求解
 - 放置於同一 `.java`
 - 使用者可重複輸入，直到輸入 ≤ 1 或回答 `N`，才結束程式
- 繳交”設計歷程”檔及 `.java`

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
```

主題：字元金字塔 - 斜金字塔

- 利用迴圈印出「*」，逐行增加印出個數，直到印出7層斜金字塔。
- 本題利用到巢狀迴圈的概念
- 巢狀迴圈為迴圈範圍內又有迴圈，從外層迴圈內層迴圈開始連層屬外層迴圈，與外層迴圈作用結束後，又回到外層迴圈。

```
public class Charstar1 {  
    public static void main(String[] args) {  
        //變數level為金字塔的層數  
        int level = 7;  
  
        //for迴圈 直到印完 level行結束金字塔  
        //外圍迴圈i為當下的層數，i增加即為換層  
        for (int i = 1; i <= level; i++) {  
            //第i列時，印出i個*  
            //j為當下的星星個數，每列都從1個開始印  
            for (int j = 1; j <= i; j++)  
                System.out.print("*");  
  
            //每層結束換行  
            System.out.println("");  
        }  
    }  
}
```

執行結果

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