Student name:_____________________ student ID:

1. Write a portion of code that defines an array of integer of size 5, then sets the array's last element to -1.

```
int[] x = new int[5];
      x[4] = -1;
```

2. Write method `public static int findFrequency(String[] a, String s)`
It returns the number of **s** occurring inside **a**. Remember that strings cannot be compared using = = . You must use "equals( )".

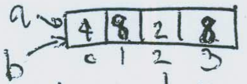
```
int count = 0;                          → มีตัวแปรไว้นับ
int length = a.length;
for (int i = 0; i < length; i++) {      → ตรวจทุกสมาชิกใน array
    if (a[i].equals(s)) count++;        → เพิ่มค่าตัวแปรอย่างถูกต้อง
}
return count;                           → return ตอนเสร็จ เมธอด
```

3. Determine the output after main is executed.

```
public static void main(String[] args) {
        int k = 1;
        int [] a = {4,1,2,8};
        f(k,a);
        System.out.println(k);   1
        showArrayContent(a);
}

public static void f(int k,int [] b){
        if (k >= b.length) return;
        for(int i=k;i<b.length;i++){
                b[i]=b[b.length-i];
        }         4     3
        k = 0;
}

public static void showArrayContent(int [] a){
        for(int i=0;i<a.length;i++) System.out.println(a[i]);
}
```



b.length = 4

```
i = 1          | i = 2          | i = 3
b[1] = b[3]    | b[2] = b[2]    | b[0] = b[1]
```

```
1
4
8
2
8
```

4. Write a method that has the following header:

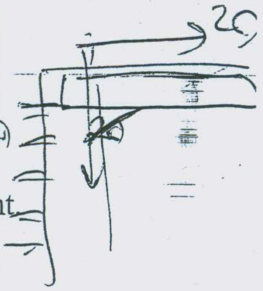
`public static int[ ] mergeTwoArray(int[] a1, int[] a2)`

This method returns a new array that is the result of alternating between a1's element and a2's element, starting from the first element of a1. For example, if a1 is {1,2,3} and a2 is {4,5,6,7,8}, the resulting array will be {1,4,2,5,3,6,7,8}.

```
int[] a3 = new int[a1.length + a2.length];
    int i1, i2, i3;
    i1 = 0; i2 = 0; i3 = 0;
while (i1 < a1.length && i2 < a2.length) {     ถ้า a1 กับ a2 ยังมีของ
    a3[i3++] = a1[i1++];                        ให้สลับเอาของใส่
    a3[i3++] = a2[i2++];
}
```

```
while (i1 < a1.length)
      a3[i3++] = a1[i1++];
while( i2 < a2.length )
      a3[i3++] = a2[i2++];
return a3;
```

อันไหนหมด ก็ทำการ อาร์เรย์ที่ยังเหลือ เอาใส่ a3 ให้หมด



5. If we want to represent seats in a cinema, we can use two-dimensional array of int. Let our cinema have 10 rows of seats, each row contains 20 seats.
   a. Declare and initialize an array that represents the seats.

```
static int[ ][ ]s = new int [10][20];
```

   b. Write method `public static void book(int row, int column)` that
      i. If the required seat has already been booked, print out a warning.
      ii. If the required seat is empty, set an array element at that position to 1.

```
if (s[row][column] == 1 )
     System.out.println("warning");
else
     s[row][column] = 1;
```

   c. Write method `public static int countEmptySeats()` that counts and returns the number of empty seats in the cinema.

```
{ int emptySeats = 0;
  for(int i=0; i<s.length; i++){
     for(int j=0; j<s[i].length ;j++){
         if( s[i][j]==0) emptySeats++; }
     }
  }
  return emptySeats;
}
```

loop ชั้นถูกต้อง

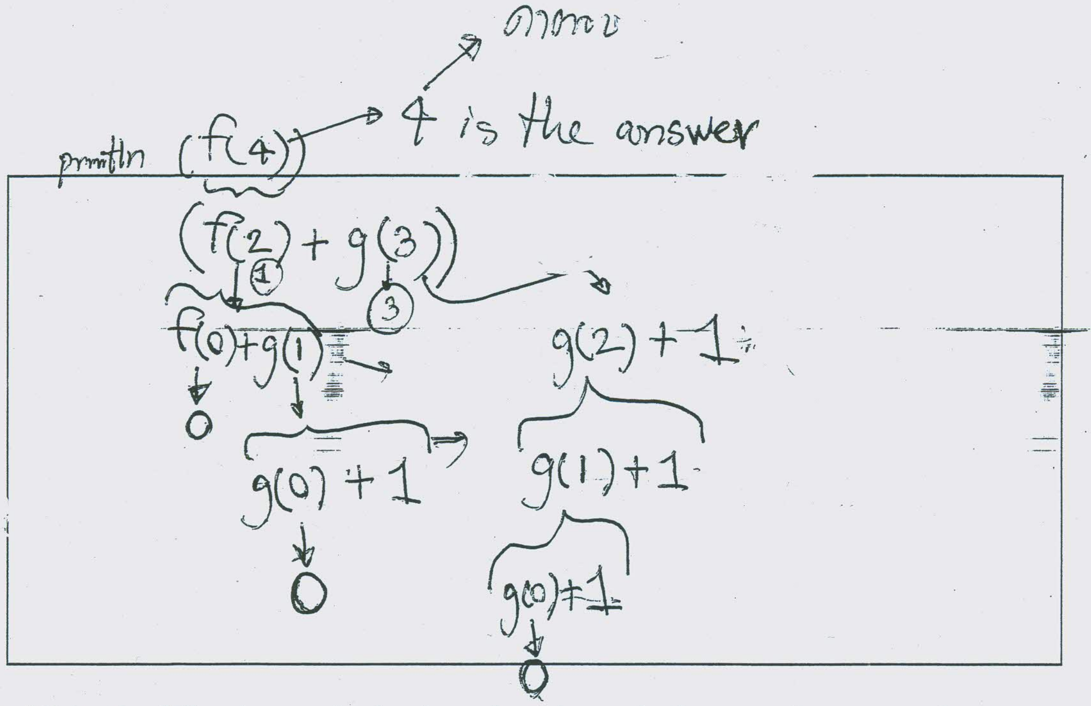
นับถูกต้อง

return ด้วย

6. What is the output when *main()* is executed? Illustrate how you come to your answer.

```
public static void main(String[] args){
        System.out.println(f(4));
}

public static int f(int n){
        if(n<=0) return 0;
        return f(n-2)+g(n-1);
}
public static int g(int n){
        if(n<=0) return 0;
        return g(n-1) +1;
}
```



7. Write the following method using a loop instead of a recursive approach.

```
public static int f(int n){
        if (n = = 0)
                return 0;
        if (n > 0)
                return (n+3) - f(n-1);
}
```

```
int result = 0
                    for(int i=1; i<=n; i++){
                        result = i+3 - result;
                    }
                    return result;
```

มีการ initialize คำตอบ

return คำตอบ

8. Using the method f from question 7, write a complete java program that prints the results of f(i), where i starts from 1 and ends at 100.

```
class A {
    public static void main(String[] args) {
            for(int i = 1; i<=100 ; i++)
                system.out.println (f(i));
    }
}
```

ต้องมี คลาส และ main

loop เพื่อปริ้น 100 ที ได้ถูกต้อง

9. Look at the following code segment. What is the value of **n** at the end of each iteration? What does this code print?

```
{       int n = 0;
        int i = 3;
        while(i >= 0){
                n = n-1;
                n = f(n);
                i--;
        }
        System.out.println(n);
}

public static int f(int n){
        return n*2;
}
```

i=3	i=2	i=1	i=0
n=-1	n=-3	n=-7	n=-15
n=f(-1) =-2	n=f(-3)=-6	n=f(-7)=-14	n=f(-15) =-30
i=2	i=1	i=0	i=-1

n is -2, -6, -14, -30

this code prints -30

10. Write a Java program that prints all the solutions of x+y+z+w =23. Each solution must be on a separate line.

```
class Program{
  public static void main(String[] args){
    for(int x=1; x<=20; x++){
      for(int y=1; y<=20; y++){
        for(int z=1; z<=20; z++){
          for(int w=1; w<=20; w++){
            if(x+y+z+w==23)   System.out.print (X); system.out.print(" ~~~~~ ") println(▲);
          }
        }
      }
    }
  }
}
```

ต้องมี class กับ main

ตัดบรรทัด
ตอนพิมพ์ตัว W
แล้ว

11. A class – Robot - only has the following code:

```
public class Robot{
        private int strength;
        public int getStrength(){return strength;}
        public void setStrength(int s){strength = s;}
}
```

Write a no argument constructor for this class. The constructor initializes strength to 5.

```
public Robot(){
    setStrength(5);
}
```

12. Write a class RobotUser. Its **main** method creates 2 Robots. The first robot has strength 10, the second robot has strength 50.

```
class RobotUser{
    public static void main(String[] args){
        Robot a = new Robot();
            a.setStrength(10);
        Robot b = new Robot();
            b.setStrength(50);
}
```

ต้องมี class เหลอ
main นี้ถูกต้อง

4

13. Write class SuperRobot that is a subclass of robot:

✓ a. A SuperRobot has an extra private variable – **flying** (it is a boolean) - that indicates whether the robot is flying.

✓ b. It has extra methods, getFlying and setFlying, similar to getStrength and SetStrength of Robot.

✓ c. It has a constructor that makes use of a no argument constructor you wrote in question 11. The constructor sets **flying** to a value given by a user.

✓ d. It has instance method `compareStrength`, which receives a Robot as an input, prints WIN if the method caller has more strength than the input, and prints LOSE otherwise.

In the main method of class SuperRobot, creates 2 Robots. The first robot being an ordinary robot (with strength = = 10) but the second being a super robot (with strength = = 20) that does not fly. Call `compareStrength` to compare the two robots.

```
public class SuperRobot extends Robot {
    private boolean flying;
    public SuperRobot (boolean fly ){
        super( );
        set Flying (fly);
    }
    public boolean getFlying (){
        return flying;
    }
    public void setFlying (boolean fly ){
        flying = fly;
    }
    public void compareStrength (Robot a ){
        if( getStrength( ) > a. getStrength( ) )
            System.out.println("WIN");
        else
            System.out.println("LOSE");
    }
    public static void main( String[ ] args) {
        Robot a = new Robot();
        a. setStrength(10);
        SuperRobot b = new SuperRobot(false);
        b. setStrength(20);
        b. compareStrength(a);
    }
```

extends ถูกต้อง

มีตัวแปร flying

เรียก super และ set flying ถูกต้อง

เขียน get กับ set ของ flying ถูกต้อง

เขียน compareStrength ถูกต้อง

initialize Robot ถูกต้อง

initialize SuperRobot ถูกต้อง

เรียก compareStrength() ด้วย super Robot