Task1. Write a nested for loop that prints the following output:

```
                        1
                    1   2   1
                1   2   4   2   1
            1   2   4   8   4   2   1
        1   2   4   8  16   8   4   2   1
    1   2   4   8  16  32  16   8   4   2   1
  1 2   4   8  16  32  64  32  16   8   4   2   1
1 2   4   8  16  32  64 128  64  32  16   8   4   2   1
```

Task2. Write a method that returns a random number from a
list of numbers passed in the argument. The method header is specified as follows: public
static int getRandom(int... numbers). Note that the program should also contain a main
method as well

Task3. The monthly payment for a given loan pays the principal and the interest. The
monthly interest is computed by multiplying the monthly interest rate and the balance
(the remaining principal). The principal paid for the month is therefore the monthly
payment minus the monthly interest.

Write a program that lets the user enter the loan amount, number of years, and interest
rate and displays the amortization schedule for the loan. Here is a sample run:

```
Loan Amount: 10000  ↵Enter
Number of Years: 1  ↵Enter
Annual Interest Rate: 7  ↵Enter

Monthly Payment: 865.26
Total Payment: 10383.21

Payment#      Interest      Principal      Balance
1             58.33         806.93         9193.07
2             53.62         811.64         8381.43
...
11            10.0          855.26         860.27
12             5.01         860.25           0.01
```

Note: The balance after the last payment may not be zero. If so, the last payment should be the normal monthly payment plus the final balance.

Hint: Write a loop to display the table. Since the monthly payment is the same for each month, it should be computed before the loop. The balance is initially the loan amount. For each iteration in the loop, compute the interest and principal, and update the balance. The loop may look like this:

for (i = 1; i <= numberOfYears * 12; i++) {

 interest = monthlyInterestRate * balance;

 principal = monthlyPayment - interest;

 balance = balance - principal;

System.out.println(i + "\t\t" + interest + "\t\t" + principal + "\t\t" + balance);

}

Task4. Zeller's congruence is an algorithm developed by Christian Zeller to calculate the day of the week. The formula is

$$h = \left(q + \frac{26(m + 1)}{10} + k + \frac{k}{4} + \frac{j}{4} + 5j\right)\%7$$

where
· h is the day of the week (0: Saturday, 1: Sunday, 2: Monday, 3: Tuesday, 4: Wednesday, 5: Thursday, 6: Friday).
· q is the day of the month.

$$h = \left(q + \frac{26(m + 1)}{10} + k + \frac{k}{4} + \frac{j}{4} + 5j\right)\%7$$

· m is the month (3: March, 4: April, ..., 12: December). January and February are counted as months 13 and 14 of the previous year.

· j is the century (i.e., $year/100$)
· k is the year of the century (i.e., year % 100).

Note that the division in the formula performs an integer division.

Write a program that prompts the user to enter a year, month, and day of the month, and displays the name of the day of the week. Here are some sample runs:

```
Enter year: (e.g., 2012): 2015 ↵Enter
Enter month: 1-12:  1 ↵Enter
Enter the day of the month: 1-31: 25 ↵Enter
Day of the week is Sunday
```

```
Enter year: (e.g., 2012): 2012 ↵Enter
Enter month: 1-12: 5 ↵Enter
Enter the day of the month: 1-31: 12 ↵Enter
Day of the week is Saturday
```

(Hint: January and February are counted as 13 and 14 in the formula, so you need to convert the user input 1 to 13 and 2 to 14 for the month and change the year to the previous year.)

Task5. Write a program that prints the following output :

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

 * * * * * * *

  * * * * *

   * * *

    *
```