Exercise: Conditional Statements Advanced

Problems for exercise and homework for the "**Programming Basics**" course [@ SoftUni Global](#).

Submit your solutions to the **SoftUni Judge** system at: https://judge.softuni.org/Contests/3692

1. Cinema

In a cinema, the chairs are arranged in a rectangular shape in **r** rows and **c** columns. There are three types of screenings with tickets at different prices:

- **`Premiere`** – premiere screening, at a price of **`12.00`** USD.
- **`Normal`** – standard screening, at a price of **`7.50`** USD.
- **`Discount`** – screening for children, and students at a reduced to a price of **`5.00`** USD.

Write a program that reads the **type of projection** (string), a **number of rows,** and a **number of columns** in the hall (integers) entered by the user and calculates the **total ticket revenue** for a **full hall**. Print the result in the format as in the examples below, **2 characters after the decimal point**.

Sample Input and Output

Input	Output
Premiere 10 12	1440.00 USD

Input	Output
Normal 21 13	2047.50 USD

Input	Output
Discount 12 30	1800.00 USD

Hints and Guidelines

1. Read the input from the console.

```
String projection = scan.nextLine();
int rows = Integer.parseInt(scan.nextLine());
int columns = Integer.parseInt(scan.nextLine());
```

2. Initialize the variable "**income**" of type "**double**" with an initial value of **0**.

```
double income = 0.0;
```

3. Perform a series of checks using the "**.equals()**" **string** comparison method, assigning the appropriate price to the "**income**" variable for each projection type ("**Premiere, Normal, Discount**") and finally printing the final result.

```
if("Premiere".equals(projection))
{
    income = rows * columns * 12;
} else if ("Normal".equals(projection)){
    income = rows * columns * 12;
}else if ("Discount".equals(projection)){
    income = rows * columns * 5;
}
System.outprintf("%.2f USD", income)
```

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3692#0

2. Summer Outfit

Summer is very changing weather, and Steven needs your help. Write a program that, **depending on the time of day and the degrees**, can recommend to Steven what clothes to wear. Your friend has different plans for each stage of the day, which require a different look, you can see them on the **table**.

2 rows are read from the console:

- **Degrees - an integer in the range [10…42]**
- **Time of the day - "Morning", "Afternoon", "Evening" (string)**

Time of the day / degrees	Morning	Afternoon	Evening
10 <= degrees <= 18	Outfit = Sweatshirt Shoes = Sneakers	Outfit = Shirt Shoes = Moccasins	Outfit = Shirt Shoes = Moccasins
18 < degrees <= 24	Outfit = Shirt Shoes = Moccasins	Outfit = T-Shirt Shoes = Sandals	Outfit = Shirt Shoes = Moccasins
degrees >= 25	Outfit = T-Shirt Shoes = Sandals	Outfit = Swim Suit Shoes = Barefoot	Outfit = Shirt Shoes = Moccasins

Print on the console: **"It's {degrees} degrees, get your {clothing} and {shoes}.""It's {degrees} degrees, get your {outfit} and {shoes}."**

Sample Input and Output

Input	Output	Comments	
16 Morning	It's 16 degrees, get your Sweatshirt and Sneakers.	In the Morning when the temperature is 16, Steven takes a sweatshirt and sneakers.	
Input	**Output**	**Input**	**Output**
22 Afternoon	It's 22 degrees, get your T-Shirt and Sandals.	28 Evening	It's 28 degrees, get your Shirt and Moccasins.

Hints and Guidelines

1. Read the input from the console, and initialize two variables "**outfit, shoes**" of type "**String**", with an initial value of empty **String** "".

```
int degrees = Integer.parseInt(scan.nextLine());
String dayTime = scan.nextLine();
String outfit = "";
String shoes = "";
```

2. Perform a series of conditions for the stage of the day, "**Morning, Afternoon, Evening**", and in the body of each statement make another condition for the degrees using a logical operator "**and**" - "**&&**", as for each range of degrees change the value of the variables "**outfit** and **shoes**" accordingly, using the table in the problem description.

```java
if ("Morning".equals(dayTime)) {
    if (degrees >= 10 && degrees <= 18) {
        outfit = "Sweatshirt";
        shoes = "Sneakers";
    } else if (degrees > 18 && degrees <= 24) {
        outfit = "Shirt";
        shoes = "Moccasins";
    } else {
        outfit = "T-Shirt";
        shoes = "Sandals";
    }
    //TODO check the other cases...
}
```

3. Print the result on the console.

```java
System.out.printf("It's %d degrees, get your %s and %s.",
        degrees, outfit, shoes);
```

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3692#1

3. New House

John and Sophie are buying a house not far from Sofia. Sophie loves flowers so much that she convinces you to write a program that will calculate **how much it will cost them**, to plant a certain number of flowers and **whether the available budget will be enough**. Different flowers have different prices.

Flowers	Roses	Dahlias	Tulips	Narcissus	Gladiolus
Price in USD	5	3.80	2.80	3	2.50

These are the following discounts:

- If Sophie buys **more than 80** Roses - **10%** discount from the final price
- If Sophie buys **more than 90** Dahlias - **15%** discount from the final price
- If Sophie buys **more than 80** Tulips - **15%** discount from the final price
- If Sophie buys **less than 120** Narcissus - the price increases by **15%**
- If Sophie Buys **less than 80** Gladiolus - the price increases by **20%**

3 rows are read from the console:

- **Type of flowers - a string with options - "Roses", "Dahlias", "Tulips", "Narcissus", "Gladiolus"**
- **Number of flowers - an integer in the range [10…1000]**
- **Budget – an integer in the range [50…2500]**

On the console print**:**

- If the budget is enough - **"Hey, you have a great garden with {number of flowers} {type of flowers} and {remaining amount} USD left."**
- If the budget is not enough - **"Not enough money, you need {needed money} USD more."**

The amount should be formatted to the second decimal place.

Sample Input and Output

Input	Output	Comments
Roses 55 250	Not enough money, you need 25.00 USD more.	Sophie wants 55 roses. The price of one rose is 5 USD, therefore for 55 pieces Sophie will have to pay: 55 * 5 = 275. However, she has a budget of 250 USD. Because 275> 250, it does not reach 25 USD.

Input	Output	Input	Output
Tulips 88 260	Hey, you have a great garden with 88 Tulips and 50.56 USD left.	Narcissus 119 360	Not enough money, you need 50.55 USD more.

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3692#2

4. Fishing Boat

Tony and his friends loved to go fishing. They were so passionate about fishing that they decided to go fishing by boat. The price of renting a boat depends on the season and the number of fishermen.

The price depends on the season:

- **The price for renting the ship in the spring is 3000 USD**
- **The price for renting the ship in summer and autumn is 4200 USD.**
- **The price for renting the ship in winter is 2600 USD.**

Depending on its number, the group receives a discount:

- **If the group is up to 6 people inclusive - a10% discount.**
- **If the group is from 7 to 11 people inclusive - a 15% discount**
- **If the group is more than 12 people - a 25% discount**

The fishermen receive an additional **5% discount if their group is an even number**. If it is **autumn** - then they do not have an additional discount.

Write a program that **calculates** whether fisherman **will have enough money**.

Input Data

3 rows are read from the console:

- **Group budget - an integer in the range [1…8000]**
- **Season - string: "Spring", "Summer", "Autumn", "Winter"**
- **Number of fishermen - an integer in the range [4…18]**

Output Data

Print on the console:

- If the **budget is enough:**
  "Yes! You have {money left} USD left."
- If the **budget is not enough**:
  "Not enough money! You need {needed money} USD."

Amounts must be formatted to two decimal places after the decimal point.

Sample Input and Output

Input	Output	Comments
3000 Summer 11	Not enough money! You need 570.00 USD.	In the Summer fishing tourism costs 4200 USD, 11 fishermen enjoy a 15% discount -> 4200 - 15% = 3570 USD, an odd number are and do not use an additional discount. 3000 <= 3570, therefore they do not reach 570.00 USD

Input	Output	Input	Output
3600 Autumn 6	Not enough money! You need 180.00 USD.	2000 Winter 13	Yes! You have 50.00 USD left.

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3692#3

5. Journey

Strangely, most people plan their vacation early. A young programmer has an **exact budget** and free time for each season. Write a program that will **accept the budget and the season**, and print, where the programmer will rest and how much he will spend.

The budget determines the **destination**, and **the season** determines how much of the budget he will spend. If it is **summer**, he will rest at the **campsite** and in the **winter** at a **hotel**. If he is in Europe, he will stay in a **hotel**, regardless of the season. Each campsite or hotel, according to the destination, has its price that corresponds to a certain **percentage of the budget**:

- At **100 USD** or less – somewhere in Bulgaria
  - **Summer** – **30%** of the budget
  - **Winter** – **70%** of the budget
- At **1000 USD** or less – somewhere in Bulgaria
  - **Summer** – **40%** of the budget
  - **Winter** – **80%** of the budget
- For more than **1000 USD**. – somewhere in Europe
  - When traveling in Europe, regardless of the season he will spend **90% of the budget**.

Input Data

2 rows are read from the console:

- The budget – a floating-point number in the range **[10.00...5000.00].**
- The season – a string "**summer**" or "**winter**"

Output Data

On the console print.

- **Somewhere in {destination}**
- **{Type of holiday} - {Money spent}**
  - The **Holiday** can be a "**Camp**", or "**Hotel**"
  - The amount must be rounded to the second decimal place

Sample Input and Output

Input	Output

50 summer	Somewhere in Bulgaria Camp - 15.00
75 winter	Somewhere in Bulgaria Hotel - 52.50
312 summer	Somewhere in Balkans Camp - 124.80
678.53 winter	Somewhere in Balkans Hotel - 542.82
1500 summer	Somewhere in Europe Hotel - 1350.00

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3692#4

6. Operations Between Numbers

Write a program that reads **two integers (N1 and N2)** and an **operator** with which to **perform** a **mathematical operation** with them. Possible operations are: **Addition (+)**, **Subtraction (-)**, **Multiplication (*)**, **Division (/)** and **Division with remainder (%)**. When adding, subtracting, and multiplying the console, the result must be printed and whether it is **even** or **odd**. In ordinary division - the result. In modular division - the remainder. It should be borne in mind that the divisor can be equal to 0 (zero) and not divisible by zero. In this case, a **specific message** must be printed.

Input Data

3 rows are read from the console:

- **N1** – **an integer in the range [0...40 000]**
- **N2** – **an integer in the range [0...40 000]**
- **Operator** – **symbol of**: `"+", "-", "*", "/", "%"`

Output Data

On the console print:

- If the operation is **addition**, **subtraction**, or **multiplication**:
  - **`"{N1} {operator} {N2} = {result} - {even/odd}"`**
- If the operation is a **division**:
  - **`"{N1} / {N2} = {result}"`** – the result is formatted to the second decimal place
- If the operation is division **by a remainder**:
  - **`"{N1} % {N2} = {remainderr}"`**
- In case of division by 0 (zero):
  - **`"Cannot divide {N1} by zero"`**

Sample Input and Output

Input	Output	Input	Output	Input	Output
10 12 +	10 + 12 = 22 - even	123 12 /	123 / 12 = 10.25	112 0 /	Cannot divide 112 by zero

10 1 -	10 - 1 = 9 - odd	10 3 %	10 % 3 = 1	10 0 %	Cannot divide 10 by zero
7 3 *	7 * 3 = 21 - odd				

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3692#5

7. Hotel Room

The hotel offers **2 types of rooms**: **studio** and **apartment**. Write a program that calculates **the price for the entire stay for a studio and apartment**. **Prices** depend on the **month** of stay:

May and October	June and September	July and August
Studio – **50** USD / night	Studio – **75.20** USD / night	Studio – **76** USD / night
Apartment – **65** USD / night	Apartment – **68.70** USD / night	Apartment – **77** USD / night

The following **discounts** are:

- **studio**, for **more than 7** nights in **May** and **October**: **5% discount**.
- **studio**, for **more than 14** nights in **May** and **October**: **30% discount**.
- **studio**, for **more than 14** nights in **June** and **September**: **20% discount**.
- **apartment**, for **more than 14** nights, **regardless of the month**: **10% discount.**

Input Data

2 rows are read from the console:

- On the first row the month – **May, June, July, August, September,** or **October**
- On the second row is the number of nights – **an integer in the range [0 ... 200]**

Output Data

On the console print:

- On the first line: **"Apartment: {price for the whole stay} USD."**
- On the second line: **"Studio: {price for the whole stay} USD."**

The price for the entire stay must be formatted to two decimal places.

Sample Input and Output

Input	Output	Comments
May 15	Apartment: 877.50 USD. Studio: 525.00 USD.	In **May**, for more than **14** nights, we reduce the price of the studio by 30% (50 - 15 = 35) and the apartment - by **10%** (65 - 6.5 = 58.5). The whole stay in the **apartment** - **877.50** USD. The whole stay in the **studio** - **525.00** USD.

Input	Output	Input	Output
June 14	Apartment: 961.80 USD. Studio: 1052.80 USD.	August 20	Apartment: 1386.00 USD. Studio: 1520.00 USD.

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3692#6

8. On Time for the Exam

A student must **go to an exam at a certain time**. He arrives at the examination room **at a certain time of arrival**. It is considered that the student **arrived on time** if he arrived at the time of the exam **or up to half an hour before**. If he arrived **more than 30 minutes earlier**, **he was early**. If he came **after** the exam time, **he was late**. Write a program that reads exam time and arrival time and records whether the student arrived on time, was early or late, and by how many hours or minutes was early or late.

Input Data

4 rows are read from the console:

- The first line **contains the exam time** - an integer from 0 to 23.
- The second line **contains the minute of the exam** – an integer from 0 to 59.
- The third line contains **the time of arrival** – an integer from 0 to 23.
- The fourth line contains **the minute of arrival** – an integer from 0 to 59.

Output Data

On the first line print:

- **"Late"** - if the student arrives later than the exam time.
- **"On time"** - if the student arrives exactly at the time of the exam or up to 30 minutes earlier.
- **"Early"** - if the student arrives more than 30 minutes before the exam time.

If the student arrives at least one minute apart from the exam time, print on the next line:

- **"mm minutes before the start"** for arriving earlier by less than an hour.
- **"hh:mm hours before the start"** or 1 hour or earlier. Always print the minutes in 2 digits, for example 1:05.
- **"mm minutes after the start"** for an hour delay.
- **"hh:mm hours after the start"** for a delay of 1 hour or more. Always print the minutes with 2 digits, for example 1:03.

Sample Input and Output

Input	Output
9 30 9 50	Late 20 minutes after the start
9 00 8 30	On time 30 minutes before the start
16 00 15 00	Early 1:00 hours before the start

Input	Output
9 00 10 30	Late 1:30 hours after the start
14 00 13 55	On time 5 minutes before the start
11 30 8 12	Early 3:18 hours before the start

Input	Output
10 00 10 00	On time
c	Early 35 minutes before the start
11 30 12 29	Late 59 minutes after the start

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3692#7

9. Ski Trip

John decided to spend his vacation in a ski resort. Before he leaves, however, he must book a hotel and **calculate how much his stay will cost**. The following types of accommodation are available, with the following accommodation prices:

- **"room for one person" – 18.00 USD per night**
- **"apartment" – 25.00 USD per night**
- **"president apartment" – 35.00 USD per night**

Depending on **the number of days** he will stay in the hotel (**example: 11 days = 10 nights**) and **the type of room he chooses**, he can enjoy different **discounts**.

The discounts are as follows:

room type	less than 10 days	between 10 and 15 days	more than 15 days
room for one person	no discount	no discount	no discount
apartment	**30%** of the final price	**35%** of the final price	**50%** of the final price
president apartment	**10%** of the final price	**15%** of the final price	**20%** of the final price

After the stay, John's rating for the hotel's services can be **positive** or **negative**. If his assessment is **positive**, John adds **25%** to the discount. If his assessment is **negative**, **10%** is deducted from the discount.

Input Data

3 rows are read from the console:

- **First row - days of stay - integer in the range [0...365]**
- **Second row – room type - "room for one person", "apartment" or "president apartment"**
- **Third row - grade - "positive" or "negative"**

Output Data

On the console print:

- **The price for his stay at the hotel is formatted to the second decimal place.**

Sample Input and Output

Input	Output	Comments
14 apartment positive	264.06	**14 days => 13 nights** => 13 * 25.00 = 325 USD **10 < 13 days < 15** => 325 – 35%= 211.25 USD **The assessment is positive** => 211.25 + 25% = 264.0625 -> 264.06 USD

Input	Output	Input	Output	Input	Output
30 president apartment negative	730.80	12 room for one person positive	247.50	2 apartment positive	21.88

Testing in the Judge System

Test the solution to this problem here: https://judge.softuni.org/Contests/Compete/Index/3692#8