**1. Cinema**

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

* **Premiere** – premiere screening, price 12.00 leva.
* **Normal** – standard screening, price 7.50 leva.
* **Discount** – discounted screening for children, pupils, and students at 5.00 leva.

Write a function that takes the type of screening (string), the number of rows, and the number of columns in the hall (integers), and calculates the total income from ticket sales if the hall is full. Print the result with **2 digits after the decimal point**.

Example input and output:

| **Input** | **Output** | **Input** | **Output** | **Input** | **Output** |
| --- | --- | --- | --- | --- | --- |
| "Premiere", 10, 12 | 1440.00 leva | "Normal", 21, 13 | 2047.50 leva | "Discount", 12, 30 | 1800.00 leva |

*Hint: use simple condition checks and arithmetic calculations. To print exactly 2 digits after the decimal point, use toFixed(2).*

**Guidelines:**

1. Initialize a variable income with a starting value of 0.
2. Use a series of conditional checks, and depending on the screening type ("Premiere", "Normal", "Discount"), calculate the correct total income, then print the result.

**2. Summer Outfit**

It’s summer with very changeable weather, and Viktor needs your help. Write a function that recommends what clothes Viktor should wear depending on the **time of day** and **temperature in degrees**. He has different plans for each part of the day, requiring a different look, shown in the table:

The function receives two arguments:

* Degrees – integer in the range [10…42]
* Text – time of day: "Morning", "Afternoon", "Evening"

| **Time of 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 one line:  
It's {degrees} degrees, get your {outfit} and {shoes}.

Example input and output:

| **Input** | **Output** |
| --- | --- |
| 16, "Morning" | It's 16 degrees, get your Sweatshirt and Sneakers. |
| 22, "Afternoon" | It's 22 degrees, get your T-Shirt and Sandals. |
| 28, "Evening" | It's 28 degrees, get your Shirt and Moccasins. |

**Guidelines:**

1. Initialize two variables: outfit, shoes.
2. Use logical conditions with && for degrees, and inside them check the time of day ("Morning", "Afternoon", "Evening"), assigning values to outfit and shoes.
3. Print the final result in the required format.

**3. New Home**

Marin and Neli are buying a house near Sofia. Neli loves flowers so much that she convinces you to write a function to calculate the cost of planting a given number of flowers and check if their budget is enough.

**Prices per flower:**

* Roses – 5 lv.
* Dahlias – 3.80 lv.
* Tulips – 2.80 lv.
* Narcissus – 3 lv.
* Gladiolus – 2.50 lv.

**Discounts/Extra costs:**

* More than 80 Roses → 10% discount.
* More than 90 Dahlias → 15% discount.
* More than 80 Tulips → 15% discount.
* Less than 120 Narcissus → price increases by 15%.
* Less than 80 Gladiolus → price increases by 20%.

The function receives 3 arguments:

* Type of flower – "Roses", "Dahlias", "Tulips", "Narcissus", "Gladiolus"
* Number of flowers – integer [10…1000]
* Budget – integer [50…2500]

Print:

* If budget is enough:  
  Hey, you have a great garden with {number} {type of flower} and {remaining money} leva left.
* If not enough:  
  Not enough money, you need {needed money} leva more.

Format the money to 2 decimal places.

**4. Fishing Boat**

Tony and his friends love fishing so much that they decide to rent a fishing boat. The rental price depends on the season and the number of fishermen.

**Season prices:**

* Spring → 3000 lv.
* Summer & Autumn → 4200 lv.
* Winter → 2600 lv.

**Discounts by group size:**

* Up to 6 people → 10% discount.
* 7–11 people → 15% discount.
* 12 or more → 25% discount.

Additionally, if the group size is even → 5% extra discount (except in Autumn).

The function receives 3 arguments:

* Budget – integer [1…8000]
* Season – "Spring", "Summer", "Autumn", "Winter"
* Number of fishermen – integer [4…18]

Print:

* If budget is enough:  
  Yes! You have {remaining money} leva left.
* Otherwise:  
  Not enough money! You need {needed money} leva.

**5. Journey**

A programmer has a budget and free time in a given season. Write a function to determine **where he will go** and **how much he will spend**.

Rules:

* Budget <= 100 lv → Bulgaria
  + Summer: spend 30% of budget (camp)
  + Winter: spend 70% of budget (hotel)
* Budget <= 1000 lv → Balkans
  + Summer: spend 40% of budget (camp)
  + Winter: spend 80% of budget (hotel)
* Budget > 1000 lv → Europe
  + Always hotel, spend 90% of budget.

Input:

* Budget – float [10.00…5000.00]
* Season – "summer" or "winter"

Output:

* Line 1: Somewhere in [Bulgaria/Balkans/Europe]
* Line 2: [Camp/Hotel] - {spent amount}

**6. Operations Between Numbers**

Write a function that receives two integers (N1 and N2) and an operator. Possible operations: +, -, \*, /, %.

* For +, -, \* → print the result and whether it is even/odd.
* For / → print the result formatted to 2 decimals.
* For % → print the remainder.
* Division by 0 → print special message.

Format:

* {N1} {op} {N2} = {result} – {even/odd}
* {N1} / {N2} = {result}
* {N1} % {N2} = {remainder}
* Cannot divide {N1} by zero

**7. Hotel Room**

A hotel offers 2 types of rooms: **studio** and **apartment**. Prices depend on the month:

| **Month** | **Studio per night** | **Apartment per night** |
| --- | --- | --- |
| May, October | 50.00 lv. | 65.00 lv. |
| June, September | 75.20 lv. | 68.70 lv. |
| July, August | 76.00 lv. | 77.00 lv. |

**Discounts:**

* Studio, >7 nights (May, Oct) → 5%
* Studio, >14 nights (May, Oct) → 30%
* Studio, >14 nights (June, Sep) → 20%
* Apartment, >14 nights (any month) → 10%

Input:

* Month – "May", "June", "July", "August", "September", "October"
* Number of nights – int [0…200]

Output:

* Apartment: {total} lv.
* Studio: {total} lv.

**8. On Time for the Exam**

A student must arrive at the exam at a given time. Depending on arrival, print whether he is **Early**, **On time**, or **Late**, and by how much.

Rules:

* On time → exactly on time or up to 30 minutes before.
* Early → more than 30 minutes before.
* Late → after the exam start.

Additional formatting:

* Less than 1 hour difference → print in minutes.
* 1+ hour difference → print h:mm format.

**9. Ski Trip**

Atanas goes skiing in Bansko. Hotel prices:

* "room for one person" – 18.00 lv/night
* "apartment" – 25.00 lv/night
* "president apartment" – 35.00 lv/night

Discounts depend on number of days (nights = days - 1):

| **Room type** | **<10 days** | **10–15 days** | **>15 days** |
| --- | --- | --- | --- |
| room for one person | no discount | no discount | no discount |
| apartment | 30% off | 35% off | 50% off |
| president apartment | 10% off | 15% off | 20% off |

After discount, apply:

* Positive review → +25%
* Negative review → -10%

Input:

* Days of stay – int [0…365]
* Room type – "room for one person", "apartment", "president apartment"
* Evaluation – "positive" or "negative"

Output:

* Final price, formatted to 2 decimals.