# Exercise: Conditional Statements Advanced

Problems for in-class and homework exercises for the course ["Programming Basics" @ SoftUni](https://softuni.org/).

**Test** your solutions in the **Judge** system: <https://judge.softuni.org/Contests/3492/Conditional-Statements-Advanced-Exercise-PS>

## Cinema

In a cinema hall, 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, priced at **12.00** USD.
* Normal – standard screening, priced at **7.50** USD.
* Discount – screening for children, schoolchildren, and students at a discounted price of **5.00** USD.

Write a function calculate(screeningType, rows, cols) - that receives a **screening type** (string), the number of **rows,** and the number of **columns** in the hall (integers) and calculates the total ticket income when the hall is full. Print the result in the format as in the examples below, rounded to 2 digits after the decimal point.

### Sample Input and Output

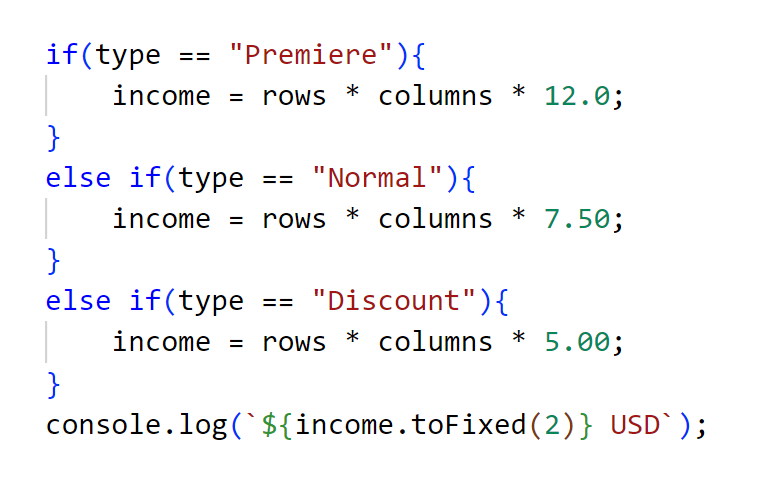
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| calculate("Premiere", 10, 12) | 1440.00 USD | calculate("Normal",  21, 13) | 2047.50 USD | calculate("Discount",12,30) | 1800.00 USD |

\* **Tip**: use simple checks and basic calculations. To output the result with exactly 2 digits after the decimal point, use **toFixed(2)**

### Hints and Guidelines

1. Convert the input numbers from string to numeric type.
2. Initialize variable "**income**" with an initial value of **0**.



1. Make a series of checks for each type of screening ("**Premiere, Normal, Discount**"), assign the corresponding price to the "**income**" variable, and at the end, print the final result.

### Testing in the Judge System

Test the solution to this problem here: [https://judge.softuni.org/Contests/Compete/Index/3492#0](https://judge.softuni.org/Contests/Compete/Index/3492#0      )

## Summer Outfit

It is a summer with very unstable weather, and Victor needs your help. Write a function to recommend what clothes Victor should wear **based on the time of day and degrees**. Your friend has different plans for each stage of the day which also require a different outfit. You can see these from the **table**.

The function calculate(degrees, timeOfDay) receives**:**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **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 the result on the console in a single line:** **"It's {degrees} degrees, get your {outfit} and {shoes}."**

### Sample Input and Output

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

### Hints and Guidelines

1. Convert the input numbers from string to numeric type.
2. Initialize two variables "**outfit, shoes**".



1. Make a check for the degrees using logical operator "**and**" – "&&" – "degrees" >= 10 && degrees <= 18" and in the body of the conditional statement for the degrees, check for the time of day "Morning, Afternoon, Evening" and for each stage of the day change the value of the variables "outfit, shoes".



1. Print the final result on the console in the format described in the problem condition.



### Testing in the Judge System

Test the solution to this problem here: [https://judge.softuni.org/Contests/Compete/Index/3492#1](https://judge.softuni.org/Contests/Compete/Index/3492#1      )

## New House

Michael and Norah buy a house not far from New York. Norah loves flowers so much that she convinces you to **write a function** to **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.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Flower** | **Rose** | **Dahlia** | **Tulip** | **Narcissus** | **Gladiola** |
| **Price per piece in USD** | 5 | 3.80 | 2.80 | 3 | 2.50 |

**The following discounts are available:**

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

The function calculate(flowersType, flowersCount, budget) receives**:**

* **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]**

**Print** on theconsole **on a single line:**

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

**Format the amount to 2 digits after the decimal point**.

### Sample Input and Output

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Output** | **Comments** | |
| calculate("Roses",  55, 250) | Not enough money, you need 25.00 USD more. | Nellie wants 55 Roses. The price of one rose is 5 USD, therefore for 55 pieces Norah will have to pay:  55 \* 5 = 275.  However, she has a budget of 250 USD.  Since 275 > 250, she needs 25 USD more. | |
| **Input** | **Output** | **Input** | **Output** |
| calculate("Tulips",  88, 260) | Hey, you have a great garden with 88 Tulips and 50.56 USD left. | calculate("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/3492#2](https://judge.softuni.org/Contests/Compete/Index/3492#2      )

## Fishing Boat

Thomas and his friends loved to go fishing. They were so passionate about fishing that they decided to go fishing by boat. The price for hiring the boat depends on **the season and the number of fishermen.**

**The price depends on the season:**

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

**Depending on their numbers, the group receives a discount:**

* **If the group is up to 6 people (inclusive) – 10% discount.**
* **If the group is from 7 to 11 people (inclusive) – 15% discount.**
* **If the group is from 12 upwards – 25% discount.**

Fishermen get an additional **5% discount if they are an even number unless it is fall - then they do not get an additional discount**.

**Write a function** to **calculate** if fishermen will **raise enough money**.

### Input Data

The function solve(budget, season, fisherman) receives:

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

### Output Data

**Print** the result on theconsole **in a single line**:

* If **the budget is enough:**

"**Yes! You have {remaining amount} USD left.**"

* If **the budget is NOT enough**:

"**Not enough money! You need {needed amount} USD.**"

**Format the amount to 2 digits after the decimal point**.

### Sample Input and Output

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Output** | **Comments** | |
| solve(3000,  "Summer",  11) | Not enough money! You need 570.00 USD. | In the summer fishing tourism costs 4200 USD, **11**  **fishermen use 15% discount -> 4200 - 15% = 3570** USD, **there are an odd number and do not use additional discount**.  3000 <= 3570, therefore they need 570.00 USD. | |
| **Input** | **Output** | **Input** | **Output** |
| solve(3600,  "Autumn",  6) | Not enough money! You need 180.00 USD. | solve(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/3492#3](https://judge.softuni.org/Contests/Compete/Index/3492#3      )

## Journey

It is strange, but most people plan their vacation in advance. A young programmer has a **certain budget** and free time in each **season**. Write a function that receives **the budget and the season** and outputs **where the programmer will be on vacation** and **how much he will spend**.

**The budget determines the destination, and the season determines how much of the budget will be spent**. If it's **summer,** he will be on a vacation at a **campsite** and in **winter at a hotel**. If it's in **Europe, regardless of the season** he will be on a vacation in a **hotel**. Each **campsite** or **hotel**, **depending on the destination**, has its **own 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 the **Balkans**
  + **Summer** – **40%** of the budget
  + **Winter** – **80%** of the budget
* At **more than 1000 USD** – somewhere in **Europe**
  + When traveling in Europe, regardless of the season, he will spend **90% of the budget**.

### Input Data

The function solve(budget, season) receives:

* **First** – Budget, **a floating-point number** in the range **[10.00...5000.00].**
* **Second** – One of the two possible seasons: **"summer"** or **"winter"**

### Output Data

**Two lines** must be printed on the console.

* **The First line** – "**Somewhere in [destination]**" among "**Bulgaria**", **"Balkans**" and "**Europe**"
* **The Second line** – "{**Vacation type**} – {**Amount spent**}"
  + **The vacation** can be among "**Camp**" and "**Hotel**"
  + **Format the amount to 2 digits after the decimal point**

### Sample Input and Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| solve(50, "summer") | Somewhere in Bulgaria  Camp - 15.00 |
| solve(75, "winter") | Somewhere in Bulgaria  Hotel - 52.50 |
| solve(312, "summer") | Somewhere in Balkans  Camp - 124.80 |
| solve("678.53, "winter") | Somewhere in Balkans  Hotel - 542.82 |
| solve("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/3492#4](https://judge.softuni.org/Contests/Compete/Index/3492#4     )

## Operations Between Numbers

Write a function solve(n1, n2, operator) that receives **two integers** (N1 and N2) and **an operator** to **perform** a **mathematical operation** on them. The possible operations are: **Addition(+)**, **Subtraction(-)**, **Multiplication(\*)**, **Division(/)** and **Modular division(%)**. When **adding**, **subtracting,** and **multiplying,** the result **must be printed on the console** and also whether it is **even** or **odd**. In an **ordinary division - the result**. In **modular division - the remainder**. It should be kept in mind that **the divisor can be equal to 0 (zero)**, and **zero is not divisible**. In this case, a **special message** must be printed.

### Input Data

The function receives **3 arguments**:

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

### Output Data

Print on theconsoleon **a single line**:

* If the operation is **Addition**, **Subtraction** or **Multiplication**:
  + "{N1} {operator} {N2} = {result} – {even/odd}"
* If the operation is **Division**:
  + "{N1} / {N2} = {result}" – the result is formatted **to 2 digits after the decimal point**
* If the operation is **Modular** **Division**:
  + "{N1} % {N2} = {remainder}"
* In case of **Division by** **0 (zero)**:
  + "Cannot divide {N1} by zero"

### Sample Input and Output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | **Output** | **Input** | **Output** | **Input** | **Output** |
| solve(10,  12,  "+") | 10 + 12 = 22 - even | solve(123,  12,  "/") | 123 / 12 = 10.25 | solve(112,  0,  "/") | Cannot divide 112 by zero |
| solve(10,  1,  "-") | 10 – 1 = 9 - odd | solve(10,  3,  "%") | 10 % 3 = 1 | solve(10,  0,  "%") | Cannot divide 10 by zero |
| solve(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/3492#5](https://judge.softuni.org/Contests/Compete/Index/3492#5      )

## Hotel Room

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

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

The following **discounts** are also available:

* For **a studio**, with **more** than **7** overnights in **May and October**: **5% discount**.
* For **a studio**, with **more** than **14** overnights in **May and October**: **30% discount**.
* For **a studio**, with **more** than **14** overnights in **June and September**: **20% discount**.
* For **an apartment**, with **more** than **14** overnights**, regardless of the month: 10% discount.**

### Input Data

The function solve(month, overnights) receives:

* **The month** – **May,** **June,** **July,** **August, September,** or **October**
* **The number of overnights –** **an integer in the range [0 … 200]**

### Output Data

**Two lines** must be **printed** on the console:

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

**Format the price for the whole stay to 2 digits after the decimal point.**

### Sample Input and Output

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Output** | **Comments** | |
| solve("May",  15) | Apartment: 877.50 USD.  Studio: 525.00 USD. | **In May**, for more than **14** overnights, 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 an **apartment** – **877.50 USD**.  The whole stay in a **studio** – **525.00 USD**. | |
| **Input** | **Output** | **Input** | **Output** |
| solve("June",  14) | Apartment: 961.80 USD.  Studio: 1052.80 USD. | solve("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/3492#6](https://judge.softuni.org/Contests/Compete/Index/3492#6      )

## On Time for the Exam

A student has to go to an **exam at a certain time** (for example, 9:30). He/she comes to the exam room at a given **arrival time** (for example, 9:40). A student is considered to be **on time** if he/she arrives at the time of the exam or up to half an hour beforehand. If he/she arrived more than 30 minutes beforehand, he/she is **early**. If he/she arrived after the exam time, he/she was **late**. Write a function that receives an exam time and an arrival time and prints whether the student arrived **on time**, whether the student was **early** or **late**, and **how many hours or minutes** late or early.

### Input Data

The function solve(examHour, examMin, arrivalHour, arrivalMin) receives:

* **The exam hour** – an integer from 0 to 23.
* **The exam minute** – an integer from 0 to 59.
* **The arrival hour** – an integer from 0 to 23.
* **The arrival minute** – 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 exam time or up to 30 minutes beforehand.
* "**Early**" if the student arrives more than 30 minutes before the exam time.

If the student arrives with at least a minute difference from the exam time, print on the next line:

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

### Sample Input and Output

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| solve(9,  30,  9,  50) | Late  20 minutes after the start | solve(9,  00,  10,  30) | Late  1:30 hours after the start | solve(10,  00,  10,  00) | On time |
| solve(9,  00,  8,  30) | On time  30 minutes before the start |  | solve(14,  00,  13,  55) | On time  5 minutes before the start | solve(11,  30,  10,  55) | Early  35 minutes before the start |
| solve(16,  00,  15,  00) | Early  1:00 hours before the start |  | solve(11,  30,  8,  12) | Early  3:18 hours before the start | solve(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/3492#7](https://judge.softuni.org/Contests/Compete/Index/3492#7      )

## Ski Trip

Albert decides to spend his vacation in Aspen and go skiing. Before he goes, however, he has to book a hotel and calculate **how much his stay will cost**. The following types of rooms are available, with the following prices per stay:

* + - **"room for one person" – 18.00 USD/per overnight**
    - "**apartment**" – **25.00 USD per overnight**
    - **"president** **apartment**" – **35.00 USD/per overnight**

Depending on **the number of days** he will stay in the hotel (**example: 11 days = 10 overnights**) and **the type of room** he will choose, he can benefit from a different **discount**.

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, Albert's feedback about the hotel's services can be **positive** or **negative**. If his feedback is positive, Albert adds **25%** of the price to the already discounted one. If his feedback is negative, he deducts 10% from the price.

### Input Data

The function solve(days, room, feedback) receives:

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

### Output Data

Print on theconsoleon **a single line**:

* **The price of his stay at the hotel, formatted to 2 digits after the decimal point.**

### Sample Input and Output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | **Output** | **Comments** | | | |
| solve(14**,**  **"**apartment**",**  **"**positive**")** | 264.06 | **14 days => 13 nights** => 13 \* 25.00 = 325 USD.  **10 < 13 days < 15** => 325 - 35% = 211.25 USD.  **The feedback is positive** => 211.25 + 25% = 264.0625 -> 264.06 USD. | | | |
| **Input** | **Output** | **Input** | **Output** | **Input** | **Output** |
| solve(30**,**  **"**president apartment**",**  **"**negative**"**) | 730.80 | solve(12**,**  **"**room for one person**",**  **"**positive**"**) | 247.50 | solve(2**,**  **"**apartment**",**  **"**positive**"**) | 21.88 |

### Testing in the Judge System

Test the solution to this problem here: [https://judge.softuni.org/Contests/Compete/Index/3492#8](https://judge.softuni.org/Contests/Compete/Index/3492#8      )