**Activity 6: Conditional Statements in Python**

Instructions: Write a Python program for each problem using the appropriate conditional statements. Ensure you apply relational and logical operators, arithmetic expressions, and input/output statements in your solutions. Save each file following this format: **Python\_Activity6.1.py**, **Python\_Activity6.2.py**, and so on, where the number corresponds to the problem number.

**1.** Write a program that asks the user to input their final grade (0-100). Based on the grade, classify it as follows:

* **90 and above** → "Excellent! You got an A."
* **80-89** → "Great job! You got a B."
* **70-79** → "Good effort! You got a C."
* **60-69** → "You passed with a D."
* **Below 60** → "You failed. Try again next time."

**2.** Ask the user to enter a number. Determine and print whether the number is:

* **Positive and even** → "The number is positive and even."
* **Positive and odd** → "The number is positive and odd."
* **Negative and even** → "The number is negative and even."
* **Negative and odd** → "The number is negative and odd."
* **Zero** → "The number is zero."

**3.** Write a program that asks the user for their **speed in km/h**. If the speed is **above 80 km/h**, calculate and display the **fine** using the formula:

**Fine = (Speed - 80) × 100**

Otherwise, print "You're driving safely."

**4.** Ask the user for the **original price of an item** and the **quantity** they want to buy. Calculate the **total cost** and apply a discount based on the total price:

* **₱5000 or more** → **20% discount**
* **₱3000 - ₱4999** → **10% discount**
* **₱1000 - ₱2999** → **5% discount**
* **Less than ₱1000** → **No discount**

Print the **final amount to be paid**.

**5.** A bank approves a loan if both conditions are met:

* The applicant's **salary is at least ₱15,000**
* The applicant’s **age is between 21 and 60**

If eligible, check the **loan amount**:

* If salary is **₱30,000 or more**, they can borrow **up to ₱500,000**.
* If salary is **between ₱15,000 and ₱29,999**, they can borrow **up to ₱200,000**.

Otherwise, print "Sorry, you do not qualify for a loan."

**Activity 6: Movie Night Snack Budget**

**Problem Statement**

A movie night snack planner helps decide what snacks to buy based on budget, number of friends, and snack preferences, offering discounts and quantity warnings if needed.  
Write a Python program named python\_activity6 that:

1. Asks the user for their total budget (in dollars), number of friends joining (whole number), snack price per pack (in dollars), and packs wanted (whole number).
2. Calculates the total snack cost using arithmetic operators: packs wanted multiplied by snack price per pack (apply PEMDAS).
3. Determines the snack plan based on these conditions:
   * If total snack cost is greater than the budget, display f'Over budget! Snack cost: ${snack\_cost:.2f}'.
   * If total snack cost is less than or equal to 25% of the budget, apply a 5% discount to the snack cost and display f'Small snack haul! Discounted cost: ${discounted\_cost:.2f}'.
   * If total snack cost is greater than 25% but less than or equal to 50% of the budget AND number of friends is 3 or more, display f'Group snack deal! Cost: ${snack\_cost:.2f}'.
   * If total snack cost is greater than 50% but less than or equal to 75% of the budget, display f'Medium snack plan! Cost: ${snack\_cost:.2f}'.
   * If total snack cost is greater than 75% of the budget but less than or equal to the budget, display f'Big snack night! Cost: ${snack\_cost:.2f}'.
4. If packs wanted multiplied by number of friends (total servings) is less than 5, display f'Warning: Might not have enough snacks!' after the cost message (for applicable cases).
5. Displays all applicable messages with costs rounded to two decimal places.