**Part 1**

**Problem 1: Personal Budget Calculator**

A lot of people find it difficult to keep track of their everyday spending. User input for income and expenses may be categorized by a Python program, which would then show monthly deficits or savings.  
Benefit: Assists users in setting savings objectives, managing money, and identifying spending trends.  
It is moderately difficult. requires knowledge of basic arithmetic and data structures, such as dictionaries or lists.

**Problem 2: Automated File Organizer**

Description: Disorganized files frequently gather on computers. Using their extensions, a Python script may group files into folders for example, MP3s into "Music," PDFs into "Documents".  
Benefit: Keeps the workspace neat and saves time.  
It is moderately difficult. involves automating operations and managing file directories.

**Part 2**

**Solve the problem using python**

Solution 1:

import datetime

import matplotlib.pyplot as plt

def get\_expenses():

expenses = {}

while True:

category = input("Enter expense category (or 'done' to finish): ").lower()

if category == "done":

break

amount = float(input(f"Enter amount for {category}: $"))

expenses[category] = expenses.get(category, 0) + amount

return expenses

def calculate\_savings(income, expenses):

total\_expenses = sum(expenses.values())

savings = income - total\_expenses

return total\_expenses, savings

def visualize\_expenses(expenses):

categories = list(expenses.keys())

amounts = list(expenses.values())

plt.pie(amounts, labels=categories, autopct='%1.1f%%')

plt.title("Expense Breakdown")

plt.show()

income = float(input("Enter your monthly income: $"))

expenses = get\_expenses()

total\_expenses, savings = calculate\_savings(income, expenses)

print(f"\nTotal Expenses: ${total\_expenses}")

print(f"Savings: ${savings}")

visualize\_expenses(expenses)

Solution 2:

import os

import shutil

def organize\_files(folder\_path):

file\_types = {

"Documents": [".pdf", ".docx", ".txt"],

"Images": [".jpg", ".jpeg", ".png"],

"Music": [".mp3", ".wav"],

"Videos": [".mp4", ".avi"],

}

for file\_name in os.listdir(folder\_path):

file\_path = os.path.join(folder\_path, file\_name)

if os.path.isfile(file\_path):

file\_ext = os.path.splitext(file\_name)[1].lower()

for folder, extensions in file\_types.items():

if file\_ext in extensions:

dest\_folder = os.path.join(folder\_path, folder)

os.makedirs(dest\_folder, exist\_ok=True)

shutil.move(file\_path, dest\_folder)

print(f"Moved: {file\_name} -> {folder}")

break

folder\_path = input("Enter the folder path to organize: ")

organize\_files(folder\_path)

**Part 3**

**Difficulty**: I chose the Budget Calculator because it seemed moderately challenging. It involved multiple Python concepts like functions, data structures, and visualization libraries.

**Challenges**: The hardest part was ensuring the visualization matched the data accurately. Debugging the integration of matplotlib was time-consuming.

**Learning**: I learned to use matplotlib effectively for creating pie charts and better understood how to manage user inputs dynamically.