asdada - sdas

dasd

sadas

Q1)

Code & Output:

Question 1

Correct

Marked out of 2.00

Flag question

Write a C program to perform ATM transaction. The transactions are Balance checking, Cash withdrawal and Cash deposition. Firstly, initialize the ATM pin as 1010 and Initial balance amount with 5000 Rupees. Take an ATM pin as input from the test case. If the input pin is equal to the initialized pin, then do the further operations. Implement a menu to do the operations like (1) Balance checking, (2) Cash withdrawal, (3) Cash deposition and (4) Quit. Use while loop to terminate or restart the process.

Input Format:

Enter your PIN

Enter your Choice # 1- Balance checking, 2-Cash withdrawal, 3-Cash deposition, 4-Quit.

Amount

Output Format

Display the balance amount

"Thanks for the Transaction" **and** For QUIT option only "Thanks for the Transaction"

```
question_images = []
code_inputs = []
output_images = []

st.write("---")
for i in range(int(number)):
    st.write(f'Question {i+1}')
    question_image = st.file_uploader(f'Upload question image for question {i+1}', type=['jpg', 'jpeg', 'png'])
    if question_image is not None:
        img = Image.open(question_image)
        img = img.convert('RGB') # convert image to RGB format
        img.save(f'temp_question_{i}.png', format='png') # save image as a PNG file
        question_images.append(f'temp_question_{i}.png')
    code_input = st.text_area(f'Paste code for question {i+1}')
    if code_input:
        code_inputs.append(code_input)
    output_image = st.file_uploader(f'Upload output image for question {i+1}', type=['jpg', 'jpeg', 'png'])
    if output_image is not None:
        img = Image.open(output_image)
        img = img.convert('RGB') # convert image to RGB format
        img.save(f'temp_output_{i}.png', format='png') # save image as a PNG file
        output_images.append(f'temp_output_{i}.png')
    st.write("---")
```

	Input	Expected	Got	
✔	1010 2 2000 4	3000 Thanks for the Transaction	3000 Thanks for the Transaction	✔
✔	1010 3 2000 2 1000 4	7000 6000 Thanks for the Transaction	7000 6000 Thanks for the Transaction	✔
✔	1010 4	Thanks for the Transaction	Thanks for the Transaction	✔

Passed all tests! ✔

Q2)

Code & Output:

Question 1

Correct

Marked out of 2.00

Flag question

Write a C program to perform ATM transaction. The transactions are Balance checking, Cash withdrawal and Cash deposition. Firstly, initialize the ATM pin as 1010 and Initial balance amount with 5000 Rupees. Take an ATM pin as input from the test case. If the input pin is equal to the initialized pin, then do the further operations. Implement a menu to do the operations like (1) Balance checking, (2) Cash withdrawal, (3) Cash deposition and (4) Quit. Use while loop to terminate or restart the process.

Input Format:

Enter your PIN

Enter your Choice # 1- Balance checking, 2-Cash withdrawal, 3-Cash deposition, 4-Quit.

Amount

Output Format

Display the balance amount

"Thanks for the Transaction" **and** For QUIT option only "Thanks for the Transaction"

```python
import streamlit as st
from fpdf import FPDF
from PIL import Image
import base64
import os
from pygments import highlight
from pygments.lexers import PythonLexer
from pygments.formatters import ImageFormatter

st.set_page_config(page_title='PDFMaker', page_icon=':memo:', layout='wide')

with st.sidebar:
    st.sidebar.image("logo.png", use_column_width=True)
    st.write("Simplifies your Moodle □□ Vtop process. Use this to simplify generating PDF's for your Moodle assignments.")
    st.warning("This app is still in development. Please report any bugs or issues.")
    st.write("Follow me on Instagram [@adarsh.py](https://www.instagram.com/adarsh.py/)!")
    st.info("Contribute here! [GitHub](https://github.com/adarshxs/PDFMaker)")
    st.info("[Compress](https://www.ilovepdf.com/compress_pdf) your generated pdf's here!", icon="□")

    st.write("")


st.title('PDF Maker')

name = st.text_input('Enter your name')
reg_num = st.text_input('Enter your registration number')
ass_name = st.text_input('Enter assignment name')
text_in = st.text_area('Enter anything you want to be printed on the first page: ')

col1, col2= st.columns(2)
with col1:
   number = st.number_input('Insert the number of questions', step=1, min_value=1)

with col2:
   theme = st.selectbox(
    'Choose a theme!',
    ('default', 'github-dark', 'sas', 'rrt', 'rainbow_dash', 'stata-light', 'gruvbox-light', 'gruvbox-light', 'monokai', 'vim', 'inkpot'))

question_images = []
code_inputs = []
output_images = []

st.write("---")
for i in range(int(number)):
    st.write(f'Question {i+1}')
    question_image = st.file_uploader(f'Upload question image for question {i+1}', type=['jpg', 'jpeg', 'png'])
    if question_image is not None:
        img = Image.open(question_image)
        img = img.convert('RGB') # convert image to RGB format
        img.save(f'temp_question_{i}.png', format='png') # save image as a PNG file
        question_images.append(f'temp_question_{i}.png')
    code_input = st.text_area(f'Paste code for question {i+1}')
    if code_input:
        code_inputs.append(code_input)
    output_image = st.file_uploader(f'Upload output image for question {i+1}', type=['jpg', 'jpeg', 'png'])
    if output_image is not None:
        img = Image.open(output_image)
        img = img.convert('RGB') # convert image to RGB format
        img.save(f'temp_output_{i}.png', format='png') # save image as a PNG file
        output_images.append(f'temp_output_{i}.png')
    st.write("---")

# set minimum line length for code inputs else shit blows up
min_line_length = 40
```

	Input	Expected	Got	
✔	1010 2 2000 4	3000 Thanks for the Transaction	3000 Thanks for the Transaction	✔
✔	1010 3 2000 2 1000 4	7000 6000 Thanks for the Transaction	7000 6000 Thanks for the Transaction	✔
✔	1010 4	Thanks for the Transaction	Thanks for the Transaction	✔

Passed all tests! ✔