

The screenshot shows a Python code editor interface with the following details:

- Title Bar:** The title bar displays "ATM.py" and includes standard window control buttons (minimize, maximize, close).
- Left Sidebar:** A vertical sidebar on the left contains several icons with corresponding status indicators:
 - File icon: Document with a blue dot.
 - Search icon: Magnifying glass.
 - Project icon: Circular with two overlapping circles.
 - Run icon: Play button.
 - Code icon: Document with a blue dot.
 - User icon: Person with a blue dot.
 - Settings icon: Gear with a blue dot.
- Code Area:** The main area contains the Python code for an ATM application. The code imports time, prints a message to insert a card, waits 5 seconds, defines a password, prompts for a pin, and initializes a balance. It then enters a loop to handle transactions based on user input (1, 2, 3, or 4). Finally, it handles a choice input and exits if option 1 is selected.
- Bottom Status Bar:** The status bar at the bottom provides information about the current file: "In 2 Col 1 Spaces: 4 LITE 8 CPU 1 Python 3.12.4 64 bit".

```
ATM.py
ATM.py > ...
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23         print("=====")
24         print("=====")
25
26     try:
27         option = int(input("Please enter your choice:"))
28
29         if option == 1:
```

ATM.py X

```
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23         print("=====")
24         print("=====")
25
26     try:
27         option = int(input("Please enter your choice:"))
28
29         if option == 1:
```

The screenshot shows a Python code editor interface with a dark theme. On the left, there's a vertical toolbar with icons for file operations, search, and other development tools. The main area displays a script named `ATM.py`. The code implements a simple ATM system with a fixed password of 1234 and an initial balance of 5000. It prompts the user to insert a card and enter a pin. If the pin is correct, it enters a loop where the user can choose between four options: 1) View balance, 2) Withdraw amount, 3) Deposit balance, or 4) Exit. The user's choice is stored in the variable `option`, which is then used to print the current balance or prompt for a withdrawal amount. The code uses standard Python syntax with print statements and if-else logic.

```
ATM.py
ATM.py > ...
2
3     print("Please insert your CARD")
4     time.sleep(5)
5
6     password = 1234
7     pin = int(input("Enter your pin:"))
8
9     print("=====")
10    print("=====")
11
12    balance = 5000
13
14    if pin == password:
15        while True:
16            print("""
17                1 == balance
18                2 == withdraw amount
19                3 == deposit balance
20                4 == exit
21            """)
22
23            print("=====")
24            print("=====")
25
26            try:
27                option = int(input("Please enter your choice:"))
28
29                if option == 1:
30                    print(f"Your current balance is {balance}")
31                    print("-----")
```

ATM.py X

```
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23         print("=====")
24         print("=====")
25
26     try:
27         option = int(input("Please enter your choice:"))
28
29         if option == 1:
```

ATM.py X

```
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5 |
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23     print("=====")
24     print("=====")
25
26 try:
27     option = int(input("Please enter your choice:"))
28
29     if option == 1:
```

ATM.py X

```
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23         print("=====")
24         print("=====")
25
26     try:
27         option = int(input("Please enter your choice:"))
28
29         if option == 1:
```

ATM.py X

```
ATM.py > [o] password
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23     print("=====")
24     print("=====")
25
26 try:
27     option = int(input("Please enter your choice:"))
28
29     if option == 1:
```

ATM.py X

```
ATM.py > [0] balance
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23         print("=====")
24         print("=====")
25
26     try:
27         option = int(input("Please enter your choice:"))
28
29         if option == 1:
```

ATM.py X

```
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8 |💡
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23         print("=====")
24         print("=====")
25
26     try:
27         option = int(input("Please enter your choice:"))
28
29         if option == 1:
```

ATM.py X

```
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23         print("=====")
24         print("=====")
25
26     try:
27         option = int(input("Please enter your choice:"))
28
29         if option == 1:
```

ATM.py X

```
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23         print("=====")
24         print("=====")
25
26     try:
27         option = int(input("Please enter your choice:"))
28
29         if option == 1:
```

The screenshot shows a Python code editor interface with a dark theme. The main window displays a script named `ATM.py`. The code implements a simple ATM system with a fixed password of 1234 and an initial balance of 5000. It includes a menu for balance inquiry, withdrawal, deposit, and exit. The code uses print statements for output and if-elif-else logic for user interaction.

```
ATM.py
ATM.py > ...
6     password = 1234
7     pin = int(input("Enter your pin:"))
8
9     print("====")
10    print("====")
11
12    balance = 5000
13
14    if pin == password:
15        while True:
16            print("""
17                1 == balance
18                2 == withdraw amount
19                3 == deposit balance
20                4 == exit
21            """)
22
23            print("====")
24            print("====")
25
26            try:
27                option = int(input("Please enter your choice:"))
28
29                if option == 1:
30                    print(f"Your current balance is {balance}")
31                    print("====")
32                    print("====")
33
34                elif option == 2:
```

The screenshot shows a Python code editor interface with a dark theme. On the left, there's a vertical toolbar with icons for file operations, search, and settings. The main area displays a script named `ATM.py`. The code implements a basic ATM system with a fixed password of 1234 and an initial balance of 5000. It includes a menu for withdrawal, deposit, and exit, and handles user input for balance retrieval and withdrawal amounts.

```
ATM.py
ATM.py > ...
6     password = 1234
7     pin = int(input("Enter your pin:"))
8
9     print("====")
10    print("====")
11
12    balance = 5000
13
14    if pin == password:
15        while True:
16            print("""
17                1 == balance
18                2 == withdraw amount
19                3 == deposit balance
20                4 == exit
21            """)
22
23            print("====")
24            print("====")
25
26            try:
27                option = int(input("Please enter your choice:"))
28
29                if option == 1:
30                    print(f"Your current balance is {balance}")
31                    print("====")
32                    print("====")
33
34                elif option == 2:
```

The screenshot shows a Python code editor interface with a dark theme. The main window displays a file named `ATM.py`. The code implements a simple ATM system with options for balance inquiry, withdrawal, deposit, and exit. The editor includes standard navigation icons (File, Edit, View, Insert, Tools, Help) and a status bar at the bottom.

```
ATM.py > ...  
16     print("")  
17     1 == balance  
18     2 == withdraw amount  
19     3 == deposit balance  
20     4 == exit  
21     "")  
22  
23     print("=====  
24     print("=====  
25  
26     try:  
27         option = int(input("Please enter your choice:"))  
28  
29         if option == 1:  
30             print(f"Your current balance is {balance}")  
31             print("=====  
32             print("=====  
33  
34         elif option == 2:  
35             withdraw_amount = int(input("Please enter withdraw amount:"))  
36  
37             if withdraw_amount > balance:  
38                 print("Insufficient funds!")  
39             else:  
40                 balance -= withdraw_amount  
41                 print(f"{withdraw_amount} is debited from your account.")  
42                 print("=====  
43                 print("=====  
44                 print(f"Your current balance is {balance}")  
In 20 Col 1/20 selected Spaces: 4 LITE 8 CR LF \ Python 3.12.4 64 bit @ Col 1
```

The screenshot shows a Jupyter Notebook interface with a dark theme. On the left, there is a sidebar with various icons: a file icon, a search icon, a link icon, a try icon (with a blue circle containing the number 1), a test tube icon, and a user icon. The main area displays a Python script named `ATM.py`. The code implements a simple ATM system with options for checking balance, withdrawing money, and depositing money. The code uses print statements for output and if-elif-else logic for handling user choices. The line 34, which starts the withdrawal logic, is highlighted with a blue background.

```
ATM.py
ATM.py > ...
4 == exit
"""
=====
print("=====")
print("=====")

try:
    option = int(input("Please enter your choice:"))

    if option == 1:
        print(f"Your current balance is {balance}")
        print("=====")
        print("=====")

    elif option == 2:
        withdraw_amount = int(input("Please enter withdraw amount:"))

        if withdraw_amount > balance:
            print("Insufficient funds!")
        else:
            balance -= withdraw_amount
            print(f"{withdraw_amount} is debited from your account.")
            print("=====")
            print("=====")
            print(f"Your current balance is {balance}")
            print("=====")
            print("=====")

    elif option == 3:
        deposit_amount = int(input("Please enter deposit amount:"))

In 25 Col 1/21 selected  Spaces: 4  LITE 8  CRIE 51 Python 3.12.4 64 bit  @ Col 1 0
```

The screenshot shows a Jupyter Notebook interface with a dark theme. On the left, there is a sidebar with various icons: a file icon, a search icon, a link icon, a try icon (with a blue circle containing the number 1), a test tube icon, and a user icon. The main area displays a Python script named `ATM.py`. The code implements a simple ATM system with options for checking balance, withdrawing money, and depositing money. The code uses print statements for output and if-elif-else logic for handling user choices. The line 34, which starts the withdrawal logic, is highlighted with a blue background.

```
ATM.py
4 == exit
"""
=====
print("=====")
print("=====")

try:
    option = int(input("Please enter your choice:"))

    if option == 1:
        print(f"Your current balance is {balance}")
        print("=====")
        print("=====")

    elif option == 2:
        withdraw_amount = int(input("Please enter withdraw amount:"))

        if withdraw_amount > balance:
            print("Insufficient funds!")
        else:
            balance -= withdraw_amount
            print(f"{withdraw_amount} is debited from your account.")
            print("=====")
            print("=====")
            print(f"Your current balance is {balance}")
            print("=====")
            print("=====")

    elif option == 3:
        deposit_amount = int(input("Please enter deposit amount:"))

In 25 Col 1/21 selected  Spaces: 4  LITE 8  CRIE 51 Python 3.12.4 64 bit  @ Col 1 0
```

ATM.py X

ATM.py > [0] withdraw_amount

```
22
23     print("=====")
24     print("=====")
25
26     try:
27         option = int(input("Please enter your choice:"))
28
29         if option == 1:
30             print(f"Your current balance is {balance}")
31             print("=====")
32             print("=====")
33
34         elif option == 2:
35             withdraw_amount = int(input("Please enter withdraw amount:"))
36
37             if withdraw_amount > balance:
38                 print("Insufficient funds!")
39             else:
40                 balance -= withdraw_amount
41                 print(f"{withdraw_amount} is debited from your account.")
42                 print("=====")
43                 print("=====")
44                 print(f"Your current balance is {balance}")
45                 print("=====")
46                 print("=====")
47
48         elif option == 3:
49             deposit_amount = int(input("Please enter deposit amount:"))
50             balance += deposit_amount
```

The screenshot shows a Jupyter Notebook interface with a dark theme. The left sidebar contains icons for file operations, search, and other notebook functions. The main area displays a Python script named `ATM.py`. The code implements a simple ATM system with four options: checking balance, withdrawing funds, depositing funds, and exiting.

```
ATM.py
ATM.py > ...
32     print("=====")
33
34     elif option == 2:
35         withdraw_amount = int(input("Please enter withdraw amount:"))
36
37         if withdraw_amount > balance:
38             print("Insufficient funds!")
39         else:
40             balance -= withdraw_amount
41             print(f"{withdraw_amount} is debited from your account.")
42             print("=====")
43             print("=====")
44             print(f"Your current balance is {balance}")
45             print("=====")
46             print("=====")
47
48     elif option == 3:
49         deposit_amount = int(input("Please enter deposit amount:"))
50         balance += deposit_amount
51         print(f"{deposit_amount} is credited to your account.")
52         print("=====")
53         print("=====")
54         print(f"Your updated balance is {balance}")
55         print("=====")
56         print("=====")
57
58     elif option == 4:
59         print("Exiting the system. Have a nice day!")
60         break
61
```

The code uses f-strings for printing messages and includes comments explaining the logic for each option. The notebook interface also shows a status bar at the bottom with various toolbars and a progress bar.

The screenshot shows a Jupyter Notebook interface with a dark theme. The left sidebar contains various icons for file operations, search, and help. The main area displays a Python script named `ATM.py`. The code implements a simple ATM system with four options: balance inquiry, withdrawal, deposit, and exit.

```
ATM.py
ATM.py > ...
33     print("=====")
34     elif option == 2:
35         withdraw_amount = int(input("Please enter withdraw amount:"))
36
37         if withdraw_amount > balance:
38             print("Insufficient funds!")
39         else:
40             balance -= withdraw_amount
41             print(f"{withdraw_amount} is debited from your account.")
42             print("=====")
43             print("=====")
44             print(f"Your current balance is {balance}")
45             print("=====")
46             print("=====")
47
48     elif option == 3:
49         balance: int = int(input("Please enter deposit amount:"))
50         balance += deposit_amount
51         print(f"{deposit_amount} is credited to your account.")
52         print("=====")
53         print("=====")
54         print(f"Your updated balance is {balance}")
55         print("=====")
56         print("=====")
57
58     elif option == 4:
59         print("Exiting the system. Have a nice day!")
60         break
61
```

The code uses f-strings for printing messages and includes comments explaining the logic for each option. The variable `balance` is defined as an integer input in the `deposit` section. The notebook interface shows the code in a single cell, with the first part of the code being executed (rows 1-47) and the second part (rows 48-61) being ready for execution. The status bar at the bottom indicates the cell is selected and provides other metadata like the kernel and Python version.

The screenshot shows a Python code editor interface with a dark theme. The main window displays a file named `ATM.py`. The code implements a simple ATM system with four options: balance inquiry, withdrawal, deposit, and exit.

```
ATM.py
ATM.py > ...
33     print("=====")
34     elif option == 2:
35         withdraw_amount = int(input("Please enter withdraw amount:"))
36
37         if withdraw_amount > balance:
38             print("Insufficient funds!")
39         else:
40             balance -= withdraw_amount
41             print(f"{withdraw_amount} is debited from your account.")
42             print("=====")
43             print("=====")
44             print(f"Your current balance is {balance}")
45             print("=====")
46             print("=====")
47
48     elif option == 3:
49         deposit_amount = int(input("Please enter deposit amount:"))
50         balance += deposit_amount
51         print(f"{deposit_amount} is credited to your account.")
52         print("=====")
53         print("=====")
54         print(f"Your updated balance is {balance}")
55         print("=====")
56         print("=====")
57
58     elif option == 4:
59         print("Exiting the system. Have a nice day!")
60         break
61
```

The code uses f-strings for printing messages. A tooltip or code completion dropdown is visible over the variable `deposit_amount` at line 49, showing its type as `int`. The editor's status bar at the bottom indicates the current line is 51, column 1, and shows other details like the file type as Python and the Python version as 3.12.4 64-bit.

The screenshot shows a Python code editor interface with a dark theme. On the left, there's a vertical toolbar with icons for file operations, search, and other development tools. The main area displays a script named `ATM.py`. The code implements a simple ATM system with options for checking balance, depositing money, withdrawing money, and exiting.

```
ATM.py
ATM.py > ...
44     print(f"Your current balance is {balance}")
45     print("=====")
46     print("=====")
47
48     elif option == 3:
49         deposit_amount = int(input("Please enter deposit amount:"))
50         balance += deposit_amount
51         print(f"{deposit_amount} is credited to your account.")
52         print("=====")
53         print("=====")
54         print(f"Your updated balance is {balance}")
55         print("=====")
56         print("=====")
57
58     elif option == 4:
59         print("Exiting the system. Have a nice day!")
60         break
61
62     else:
63         print("Invalid option. Please try again.")
64         print("=====")
65         print("=====")
66
67     except ValueError:
68         print("Invalid input. Please enter a numeric value.")
69         print("=====")
70         print("=====")
71
72     else:
```

The screenshot shows a code editor interface with a dark theme. On the left, there's a vertical toolbar with icons for file operations, search, and other development tools. The main area displays a Python script named `ATM.py`. The script handles user input for an ATM system, including depositing money, exiting, and checking for a valid pin. It includes error handling for non-numeric inputs.

```
ATM.py
ATM.py > ...

48     elif option == 3:
49         deposit_amount = int(input("Please enter deposit amount:"))
50         balance += deposit_amount
51         print(f"{deposit_amount} is credited to your account.")
52         print("====")
53         print("====")
54         print(f"Your updated balance is {balance}")
55         print("====")
56         print("====")
57
58     elif option == 4:
59         print("Exiting the system. Have a nice day!")
60         break
61
62     else:
63         print("Invalid option. Please try again.")
64         print("====")
65         print("====")
66
67     except ValueError:
68         print("Invalid input. Please enter a numeric value.")
69         print("====")
70         print("====")
71
72     else:
73         print("Wrong pin, please try again with the correct pin :(")
74
```

At the bottom, status bar indicators show the current line (Ln 62), column (Col 21 / 7 selected), spaces used (Spaces: 4), LITE 8, CRIF, Python 3.12.4 64 bit, and Go Live.

ATM.py X

ATM.py > ...

```
54         print(f"Your updated balance is {balance}")
55         print("=====")
56         print("=====")
57
58     elif option == 4:
59         print("Exiting the system. Have a nice day!")
60         break
61
62     else:
63         print("Invalid option. Please try again.")
64         print("=====")
65         print("=====")
66
67     except ValueError:
68         print("Invalid input. Please enter a numeric value.")
69         print("=====")
70         print("=====")
71
72 else:
73     print("Wrong pin, please try again with the correct pin :(")
74
```

ATM.py X

ATM.py > ...

```
54         print(f"Your updated balance is {balance}")
55         print("=====")
56         print("=====")
57
58     elif option == 4:
59         print("Exiting the system. Have a nice day!")
60         break
61
62     else:
63         print("Invalid option. Please try again.")
64         print("=====")
65         print("=====")
66
67     except ValueError:
68         print("Invalid input. Please enter a numeric value.")
69         print("=====")
70         print("=====")
71
72 else:
73     print("Wrong pin, please try again with the correct pin :(")
74
```

▷ ⌂ ⌂ ⌂



ATM.py X

ATM.py > ...

```
54         print(f"Your updated balance is {balance}")
55         print("=====")
56         print("=====")
57
58     elif option == 4:
59         print("Exiting the system. Have a nice day!")
60         break
61
62     else:
63         print("Invalid option. Please try again.")
64         print("=====")
65         print("=====")
66
67     except ValueError:
68         print("Invalid input. Please enter a numeric value.")
69         print("=====")
70         print("=====")
71
72 else:
73     print("Wrong pin, please try again with the correct pin :(")
74
```

ATM.py X

ATM.py > ...

```
54         print(f"Your updated balance is {balance}")
55         print("=====")
56         print("=====")
57
58     elif option == 4:
59         print("Exiting the system. Have a nice day!")
60         break
61
62     else:
63         print("Invalid option. Please try again.")
64         print("=====")
65         print("=====")
66
67     except ValueError:
68         print("Invalid input. Please enter a numeric value.")
69         print("=====")
70         print("=====")
71
72 else:
73     print("Wrong pin, please try again with the correct pin :(")
74
```

A screenshot of a Python code editor interface. The main area displays a script named `ATM.py` with the following content:

```
1 import time
2
3 print("Please insert your CARD")
4 time.sleep(5)
5
6 password = 1234
7 pin = int(input("Enter your pin:"))
8
9 print("=====")
10 print("=====")
11
12 balance = 5000
13
14 if pin == password:
15     while True:
16         print("""
17             1 == balance
18             2 == withdraw amount
19             3 == deposit balance
20             4 == exit
21             """)
22
23     print("=====")
24     print("=====")
25
26 try:
27     option = int(input("Please enter your choice:"))
28
29     if option == 1:
```

The code implements a simple ATM machine simulation. It prompts the user to insert a card and waits for 5 seconds. It then asks for the PIN and checks it against the stored password (1234). If the PIN is correct, a menu is displayed with options 1 through 4. Option 1 shows the current balance. Options 2 and 3 allow the user to withdraw or deposit money, respectively. Option 4 exits the loop. The program then prints two separator lines and enters a `try` block to handle user input for the choice.

```
PS C:\Users\caysu\Onedrive\Desktop\Python> python ATM.py  
Please insert your CARD
```

```
PS C:\Users\caysu\Onedrive\Desktop\Python> python ATM.py
```

```
Please insert your CARD
```

```
Enter your pin:1234
```

```
=====
```

```
=====
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====
```

```
=====
```

```
Please enter your choice:
```

```
PS C:\Users\caysu\Onedrive\Desktop\Python> python ATM.py
```

```
Please insert your CARD
```

```
Enter your pin:1234
```

```
=====
```

```
=====
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====
```

```
=====
```

```
Please enter your choice:1
```

```
Your current balance is 5000
```

```
=====
```

```
=====
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====
```

```
=====
```

```
Please enter your choice:
```

```
4 == exit
```

```
=====  
=====  
Please enter your choice:1  
Your current balance is 5000  
=====
```

```
=====  
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====  
=====  
Please enter your choice:2  
Please enter withdraw amount:1000  
1000 is debited from your account.  
=====
```

```
=====  
=====  
Your current balance is 4000  
=====
```

```
=====  
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====  
=====  
Please enter your choice:
```

```
4 == exit
```

```
=====  
=====  
Please enter your choice:1  
Your current balance is 5000  
=====
```

```
=====  
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====  
=====  
Please enter your choice:2  
Please enter withdraw amount:1000  
1000 is debited from your account.  
=====
```

```
=====  
=====  
Your current balance is 4000  
=====
```

```
=====  
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====  
=====  
Please enter your choice:
```

```
Please enter your choice:2  
Please enter withdraw amount:1000  
1000 is debited from your account.
```

```
=====
```

```
=====
```

```
Your current balance is 4000
```

```
=====
```

```
=====
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====
```

```
=====
```

```
Please enter your choice:3  
Please enter deposit amount:500  
500 is credited to your account.
```

```
=====
```

```
=====
```

```
Your updated balance is 4500
```

```
=====
```

```
=====
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====
```

```
=====
```

```
Please enter your choice:
```

```
1000 is debited from your account.  
=====
```

```
=====  
Your current balance is 4000  
=====
```

```
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
Please enter your choice:3  
Please enter deposit amount:500  
500 is credited to your account.  
=====
```

```
=====  
Your updated balance is 4500  
=====
```

```
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
=====  
Please enter your choice:4  
Exiting the system. Have a nice day!  
PS C:\Users\caysu\Onedrive\Desktop\Python>
```

```
=====  
Your current balance is 4000  
=====
```

```
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
=====  
Please enter your choice:3  
Please enter deposit amount:500  
500 is credited to your account.  
=====
```

```
=====  
Your updated balance is 4500  
=====
```

```
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
=====  
Please enter your choice:4  
Exiting the system. Have a nice day!  
PS C:\Users\caysu\Onedrive\Desktop\Python> python ATM.py  
Please insert your CARD  
Enter your pin:|
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
Please enter your choice:3  
Please enter deposit amount:500  
500 is credited to your account.
```

```
Your updated balance is 4500
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
Please enter your choice:4  
Exiting the system. Have a nice day!  
PS C:\Users\caysu\Onedrive\Desktop\Python> python ATM.py  
Please insert your CARD  
Enter your pin:1245
```

```
Wrong pin, please try again with the correct pin :(
```

```
PS C:\Users\caysu\Onedrive\Desktop\Python> |
```

```
Your updated balance is 4500
```

```
=====
```

```
=====
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====
```

```
=====
```

```
Please enter your choice:4
```

```
Exiting the system. Have a nice day!
```

```
PS C:\Users\caysu\Onedrive\Desktop\Python> python ATM.py
```

```
Please insert your CARD
```

```
Enter your pin:1245
```

```
=====
```

```
=====
```

```
Wrong pin, please try again with the correct pin :(
```

```
PS C:\Users\caysu\Onedrive\Desktop\Python> python ATM.py
```

```
Please insert your CARD
```

```
Enter your pin:1234
```

```
=====
```

```
=====
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====
```

```
=====
```

```
Please enter your choice:|
```

```
4 == exit
```

```
=====  
=====  
Please enter your choice:1  
Your current balance is 5000  
=====
```

```
=====  
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====  
=====  
Please enter your choice:2  
Please enter withdraw amount:5000  
5000 is debited from your account.  
=====
```

```
=====  
=====  
Your current balance is 0  
=====
```

```
=====  
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit
```

```
=====  
=====  
Please enter your choice:\|
```

```
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
Please enter your choice:1  
Your current balance is 0  
=====
```

```
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
Please enter your choice:2  
Please enter withdraw amount:1000  
Insufficient funds!  
=====
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
Please enter your choice:|
```

```
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
Please enter your choice:1  
Your current balance is 0  
=====
```

```
=====  
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
Please enter your choice:2  
Please enter withdraw amount:1000  
Insufficient funds!  
=====
```

```
1 == balance  
2 == withdraw amount  
3 == deposit balance  
4 == exit  
=====
```

```
Please enter your choice:|
```