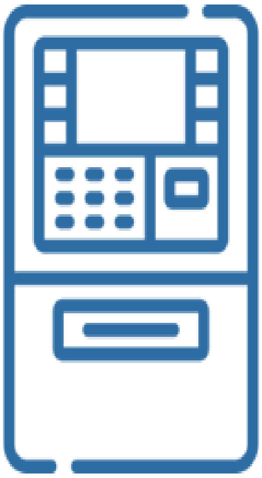
EXERCICE 1

ATM

To facilitate grading and peer testing, it is essential that you strictly follow the instructions provided:

- **Only program what is explicitly requested.**
- **Do not add anything beyond what is requested.**
- **Do not perform tasks in a manner different from what is instructed.**



For this programming exercise, you will write a program that simulates, in a simplified manner, the operations available with an ATM: depositing money, withdrawing money, and checking the account balance:

1. The account balance inquiry operation allows you to know the amount of money available.
2. The deposit operation allows you to specify the amount of cash deposited, either in CHF or EUR, calculate the new account balance, and display it.
3. The withdrawal operation allows you to specify the amount of cash to withdraw, either in CHF or EUR. It should check if the withdrawal is possible, calculate the distribution of the amount into bill denominations, update the account balance, and display it.

The program should begin by displaying the menu of available ATM operations:

```
Choose your operation:
    1)  Deposit
    2)  Withdrawal
    3)  Account consultation
    4)  Finish
Your choice >
```

Each time an operation is selected, it must be executed, and then the menu of available operations should be displayed again until the customer selects option 4 to finish his/her transactions on the bank account.

After choosing the **first** operation, the program should start by verifying the account's PIN code (unless the operation is option 4, which must be executed normally).

```
Enter your pin code >
```

The entry of the PIN code is repeated as long as the customer makes a mistake. With each error, a message is displayed indicating that there is an error, and the code needs to be entered again:

```
"Invalid PIN code, you have ... attempts remaining."
```

After 3 unsuccessful attempts, the program stops after displaying the message:

```
"For your protection, banking operations will be suspended, please retrieve your card."
```

The PIN code does not need to be repeated for subsequent operations. It only occurs for the first operation.

Instruction for the PIN code: The variable containing the PIN code is a String that contains the value "INTRO1234" (you must use exactly this PIN code value in your program).

Deposit operation

If the deposit operation is chosen, the customer must respond to several questions:

Indicate the currency of the deposit: 1) CHF; 2) EUR >

Then:

Indicate the amount of the deposit >

Code instructions: The deposit amount variable is an Int. The amount available on the account is in CHF. Transactions in EUR must be converted to CHF.

The smallest denominations accepted for cash deposits are 10 CHF or 10 EUR. The program must therefore check that the deposit amount is divisible by 10. If the amount is not a multiple of 10, the program must display the message:

The amount must be a multiple of 10.

In this case, the amount must be entered again.

The program should then update the bank account amount, after converting the EUR amount to CHF if necessary (using the following fixed rate: 1 EUR is worth 0.95 CHF et 1 CHF is worth 1.05 EUR - you must use exactly this conversion rate amount in your program) and display the new amount with the following message:

Your deposit has been processed, the new amount available on your account is: …

Code Instructions: The variable for the account balance is a type Double. It must be initialized in the program with the initial value of 1200.0 (you must use exactly this initial account balance value).

Consultation operation

If the consultation operation is selected, the program displays the amount available in the bank account:

The amount available on your account is: …

Withdrawal operation

If the withdrawal operation is chosen, the customer must answer several questions:

Indicate the currency: 1) CHF; 2) EUR >

The message should repeat as long as the customer makes a choice other than 1 or 2.

Then:

Indicate the amount of the withdrawal.

As with the deposit, the amount must be a multiple of 10. The entry is repeated until the amount respects this constraint by indicating a message to the customer:

The amount must be a multiple of 10.

The amount must also be less than or equal to the authorized withdrawal amount. This amount corresponds to 10% of the total amount available on the account. The entry must also be repeated until the amount respects this constraint, indicating an error message for the customer:

Your authorized withdrawal limit is: …

Then, if the currency is CHF:

In 1) large denominations, 2) small denominations>

The message should repeat as long as the user makes a choice other than 1 or 2.

The programme should check whether the amount can be divided into large denominations (it must be greater than or equal to 200). If it cannot, the choice of large or small denominations should not be offered. It will necessarily be in small denominations.

For CHF withdrawals, the denominations available are 500, 200, 100, 50, 20 and 10. If withdrawal is requested in large denominations, the denominations used start from 500. If withdrawal is requested in small denominations, the denominations used start from 100.

For withdrawal in EUR, the denominations available are 100, 50, 20 and 10. There is no choice between large and small denominations.

The customer then receives the requested amount divided into different denominations.

The program will assist the customer in indicating the number of each bill denomination they want, ensuring that the total matches the requested total amount. The program will start systematically with the largest bill denomination and proceed to the smallest. For each bill denomination, the program must determine the maximum possible quantity. If this quantity is 0, the program should not suggest that bill denomination.

For example, if the withdrawal amount is 650 CHF in large bill denominations, the customer will experience the following sequence of displays and input requests:

There is 650 CHF left to distribute.
You can get a maximum of 1 bill(s) of 500 CHF.
Type 'o' for OK or any other value less than the proposed one:

At each step, the program must ensure that if a value is entered, it must be strictly less than the one offered. The program should repeat the input request until this condition is met or until the customer enters 'o'.

If the customer enters the value 'o', he/she will receive the following display:

There is 150 CHF left to distribute.
You can get a maximum of 1 bill(s) of 100 CHF.

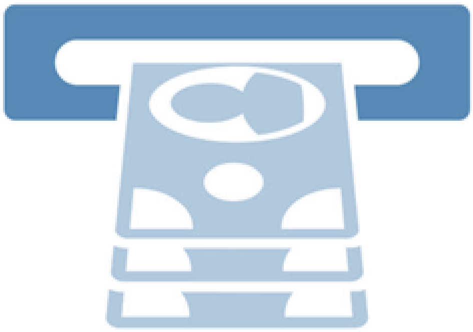
Type 'o' for OK or any other value less than the proposed one.

If the customer enters the value 0, he/she will receive the following display:

There is 150 CHF left to distribute.
You can get a maximum of 3 bill(s) of 50 CHF.
Type 'o' for OK or any other value less than the proposed one.

If the customer enters the value 'o', he/she will receive the following display:

Please withdraw the requested amount:
1 bill(s) of 500 CHF
3 bill(s) of 50 CHF



For the same withdrawal amount in large bill denominations, the distribution may vary as follows:

There is 650 CHF left to distribute.
You can get a maximum of 1 bill(s) of 500 CHF.
Type 'o' for OK or any other value less than the proposed one.

If the customer enters the value 0:

There is 650 CHF left to distribute.
You can get a maximum of 3 bill(s) of 200 CHF.
Type 'o' for OK or any other value less than the proposed one.

If the customer enters the value 2:

There is 250 CHF left to distribute.
You can get a maximum of 2 bill(s) of 100 CHF.
Type 'o' for OK or any other value less than the proposed one.

If the customer enters 'o':
There is 50 CHF left to distribute.

You can get a maximum of 1 bill(s) of 50 CHF.
Type 'o' for OK or any other value less than the proposed one.

If the customer enters 'o':

Please withdraw the requested amount:
2 bill(s) of 200 CHF
2 bill(s) of 100 CHF
1 bill(s) of 50 CHF

The undistributed bill denominations, here for example, 500, 20, and 10, must not be displayed (there should be no display like "0 bill(s) of 500").

The program should update the bank account amount, after converting the EUR amount to CHF if necessary (using the following fixed rate: 1 EUR is worth 0.95 CHF et 1 CHF is worth 1.05 EUR - you should use exactly this conversion rate amount in your program) and display the new amount with the following message:

Your withdrawal has been processed, the new amount available on your account is: …

Code instructions: Use integer division to determine the largest bill denomination value that can be given based on the total amount.

Important Note: In the special situation where the customer hasn't selected any bill denominations before reaching the distribution of 10 CHF or 10 EUR (i.e., if they have chosen the value 0 for all bill denominations before reaching 10 CHF or 10 EUR), they will be required to accept the program's proposal without the option to modify it. Otherwise, the distributed amount would be smaller than the requested amount, which would not be consistent.

Code instructions: For displaying the account balance, it's expected to display values rounded to 2 decimal places for cents. One possible method is to use the `printf` method for value displays in the following way.

The code fragment below:

```
printf("The one-digit value of Pi is: %.1f \n", math.Pi)

printf("The two-digit value of Pi is: %.2f \n", math.Pi)

printf("The three-digit value of Pi is: %.3f \n", math.Pi)

printf("The four-digit value of Pi is: %.4f \n", math.Pi)
```

```
print(math.Pi)
```

produces the following result:

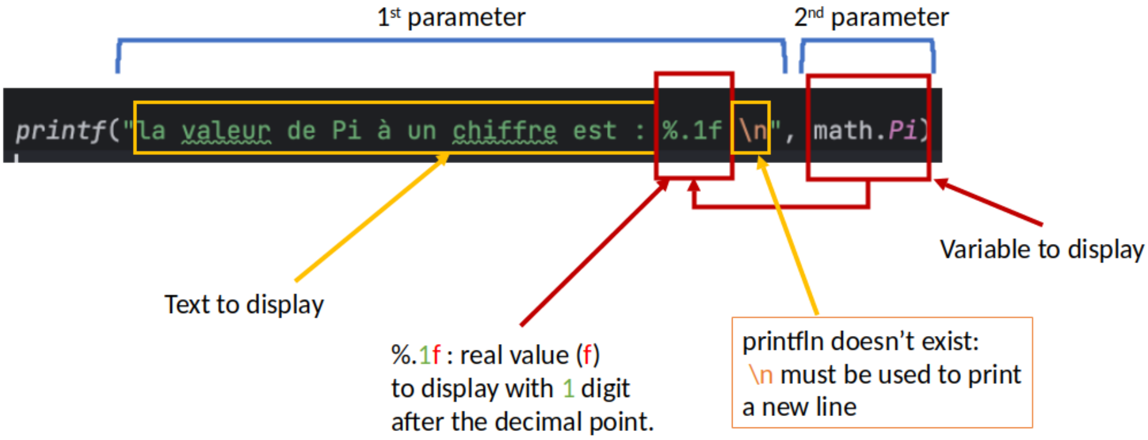
```
The one-digit value of Pi is: 3.1

The two-digit value of Pi is: 3.14

The three-digit value of Pi is: 3.142

The four-digit value of Pi is: 3.1416

3.141592653589793
```

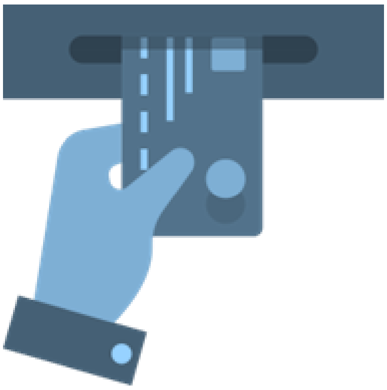


```
printf("la valeur de Pi à un chiffre est : %.1f \n", math.Pi)
```

1st parameter
2nd parameter
Text to display
%.1f : real value (f) to display with 1 digit after the decimal point.
printfln doesn't exist: \n must be used to print a new line
Variable to display

Use the `printf` method to display the account balance.

End operation



When the customer chooses to finish the operations, the program ends after displaying the message:

End of operations, don't forget to collect your card.

Test example

Deposit Operation:

1. If the account balance is at its initial value, 1200.0 CHF, depositing 120 CHF will increase the account balance to 1320.0 CHF.
2. If the deposit is 120 EUR, the account balance will become 1314.0 CHF.
3. If the account balance is at the value of 2010.0 CHF, depositing 350 CHF will increase the account balance to 2360.0 CHF.
4. If the deposit is 350 EUR, the account balance will become 2342.5 CHF.

Withdrawal Operation:

1. If the account balance is at its initial value, 1200.0 CHF, withdrawing 120 CHF will result in smaller denominations. It can be distributed as 1 bill of 100 CHF and 1 bill of 20 CHF. The account balance will decrease to 1080.0 CHF.
2. If the withdrawal is 120 EUR, the customer can receive the amount distributed as 1 bill of 100 EUR and 1 bill of 20 EUR. The account balance will decrease to 1086.0 CHF.
3. If the account balance is at the value of 7000.0 CHF, withdrawing 610 CHF in large denominations can be distributed as 1 bill of 500 CHF, 1 bill of 100 CHF, and 1 bill of 10 CHF. In smaller denominations, it can be distributed as 6 bills of 100 CHF and 1 bill of 10 CHF. The account balance will decrease to 6390.0 CHF.
4. If the withdrawal is 610 EUR, the distribution can be as 5 bills of 100 EUR, 2 bills of 50 EUR, and 1 bill of 10 EUR. The account balance will decrease to 6420.0 CHF.