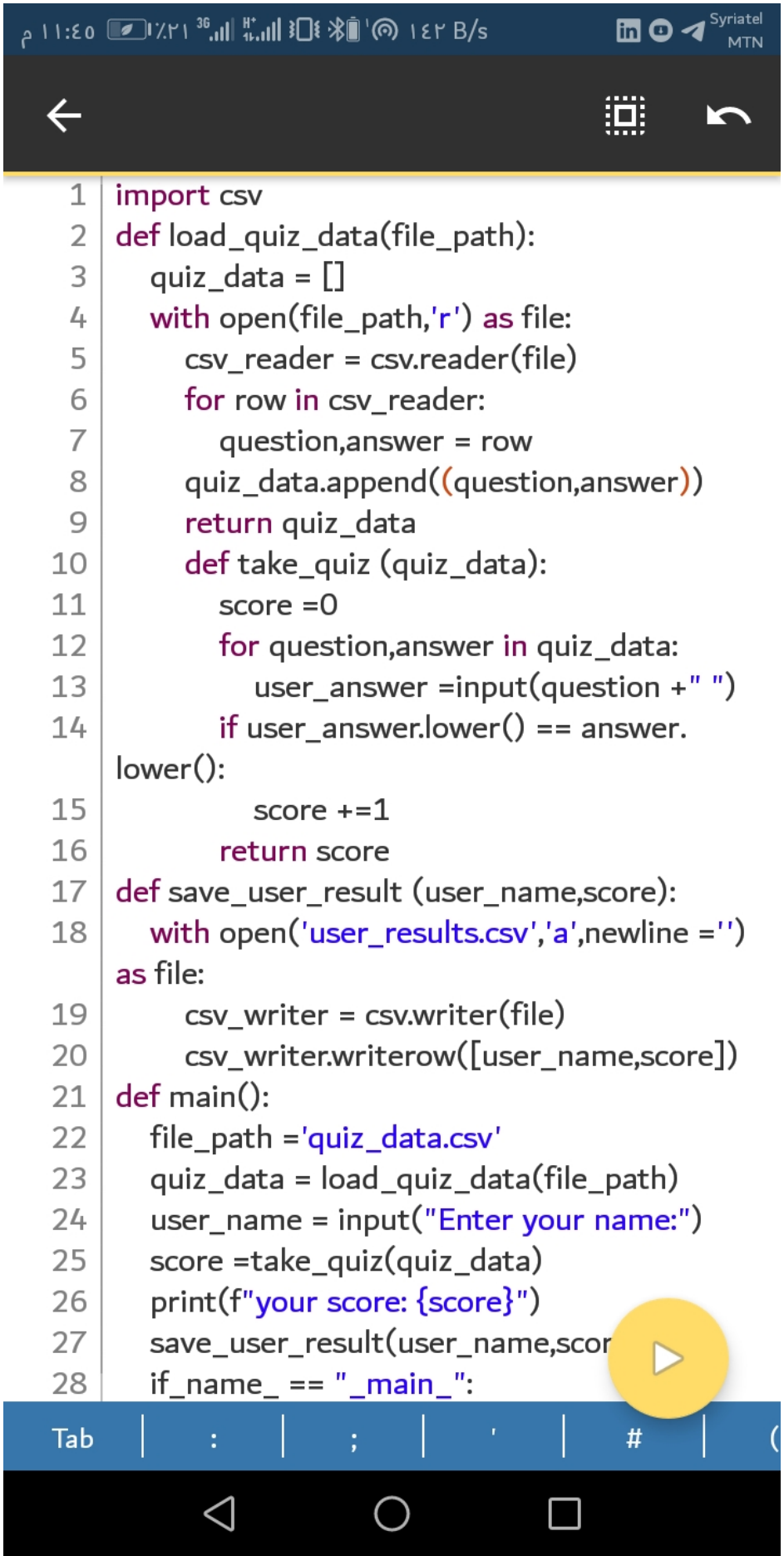
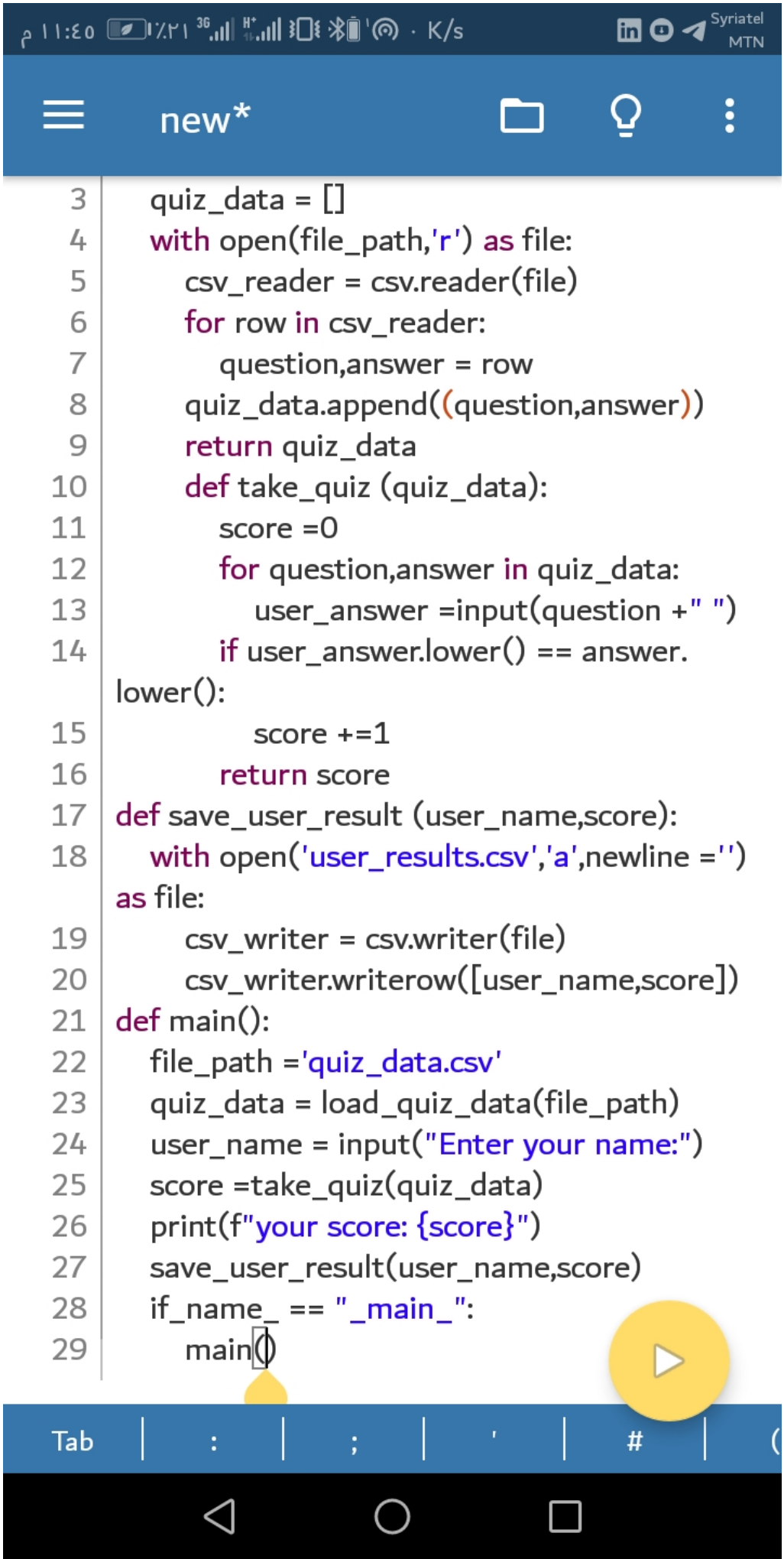
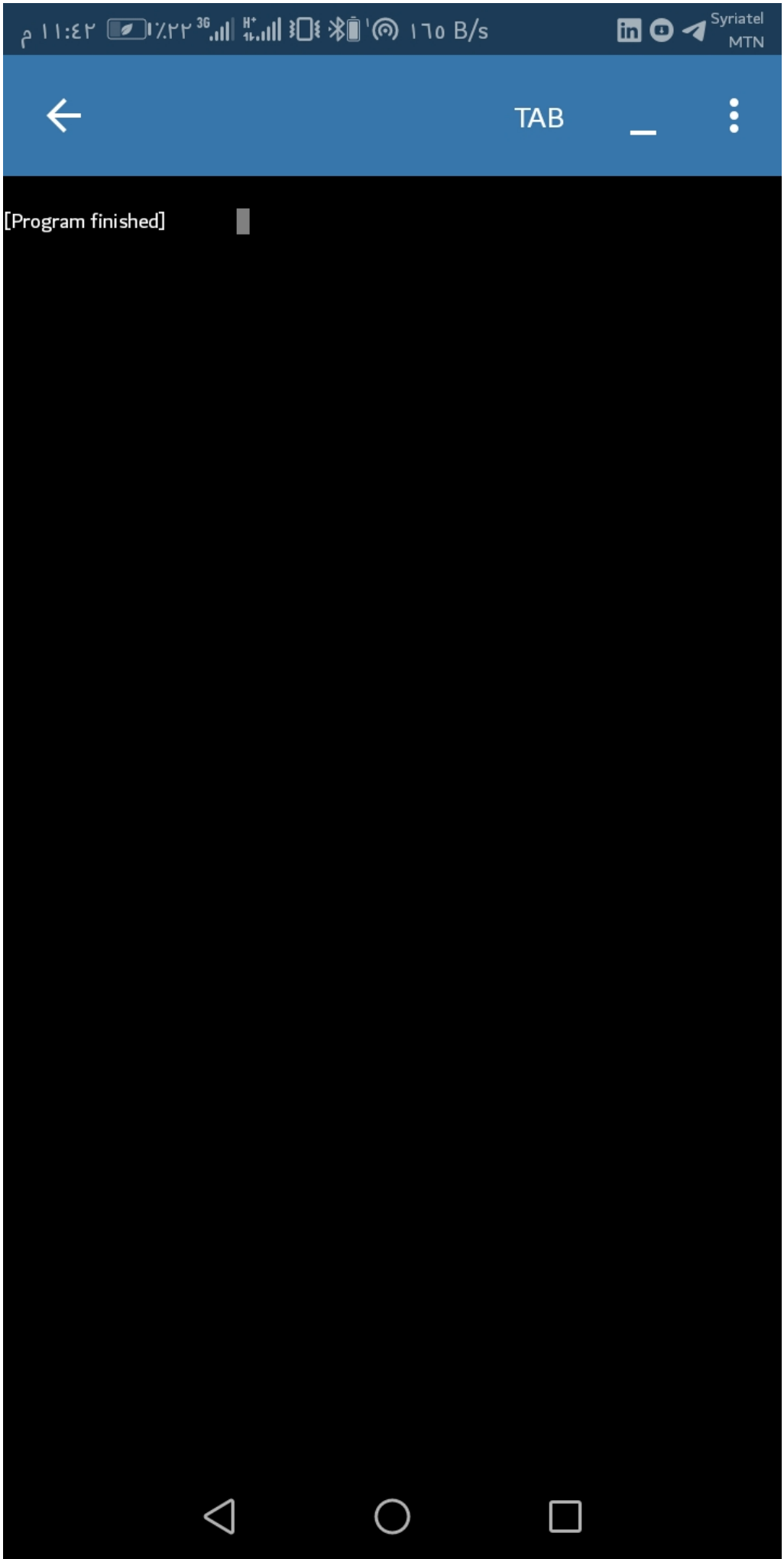
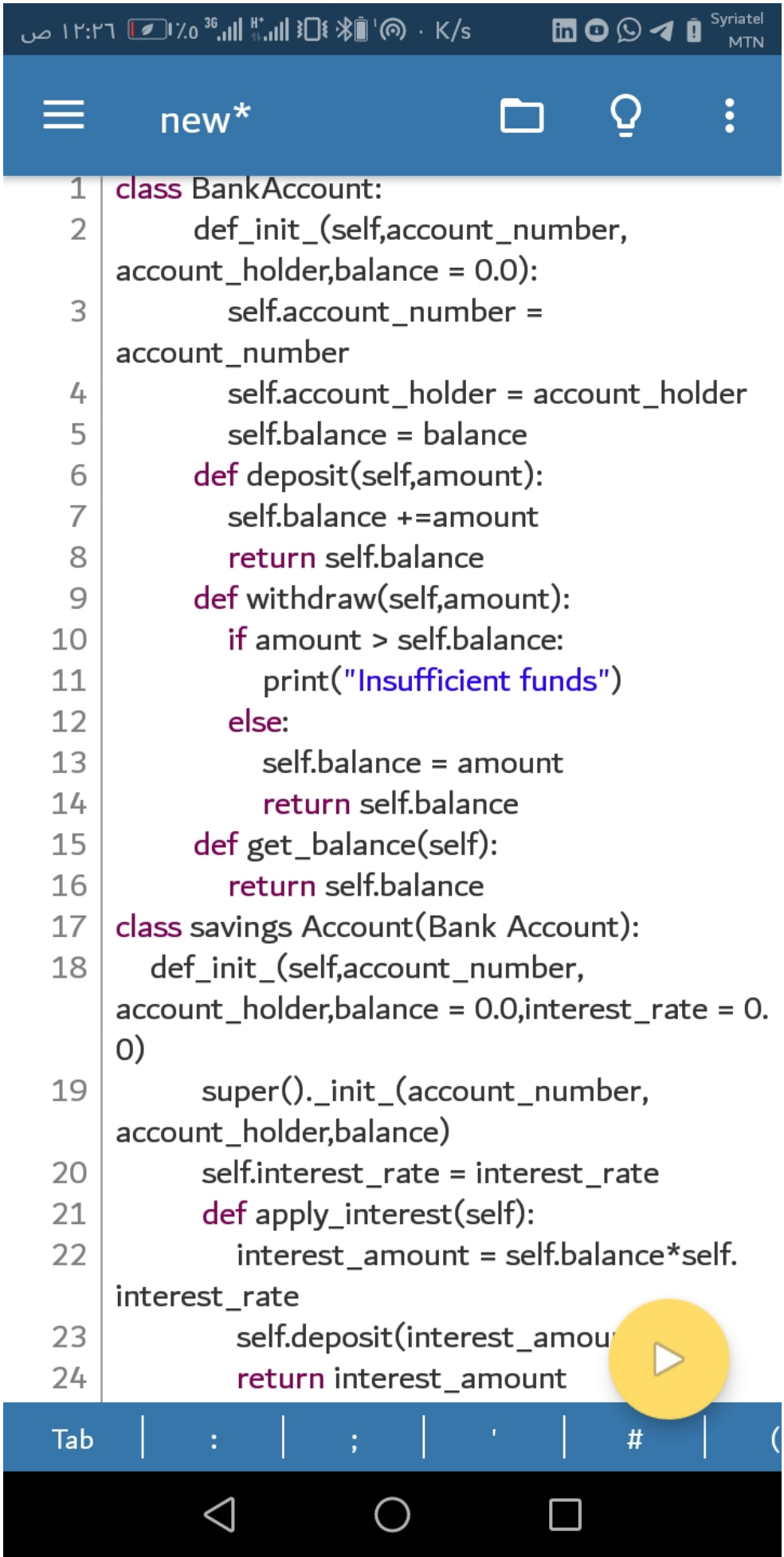
:Question 3

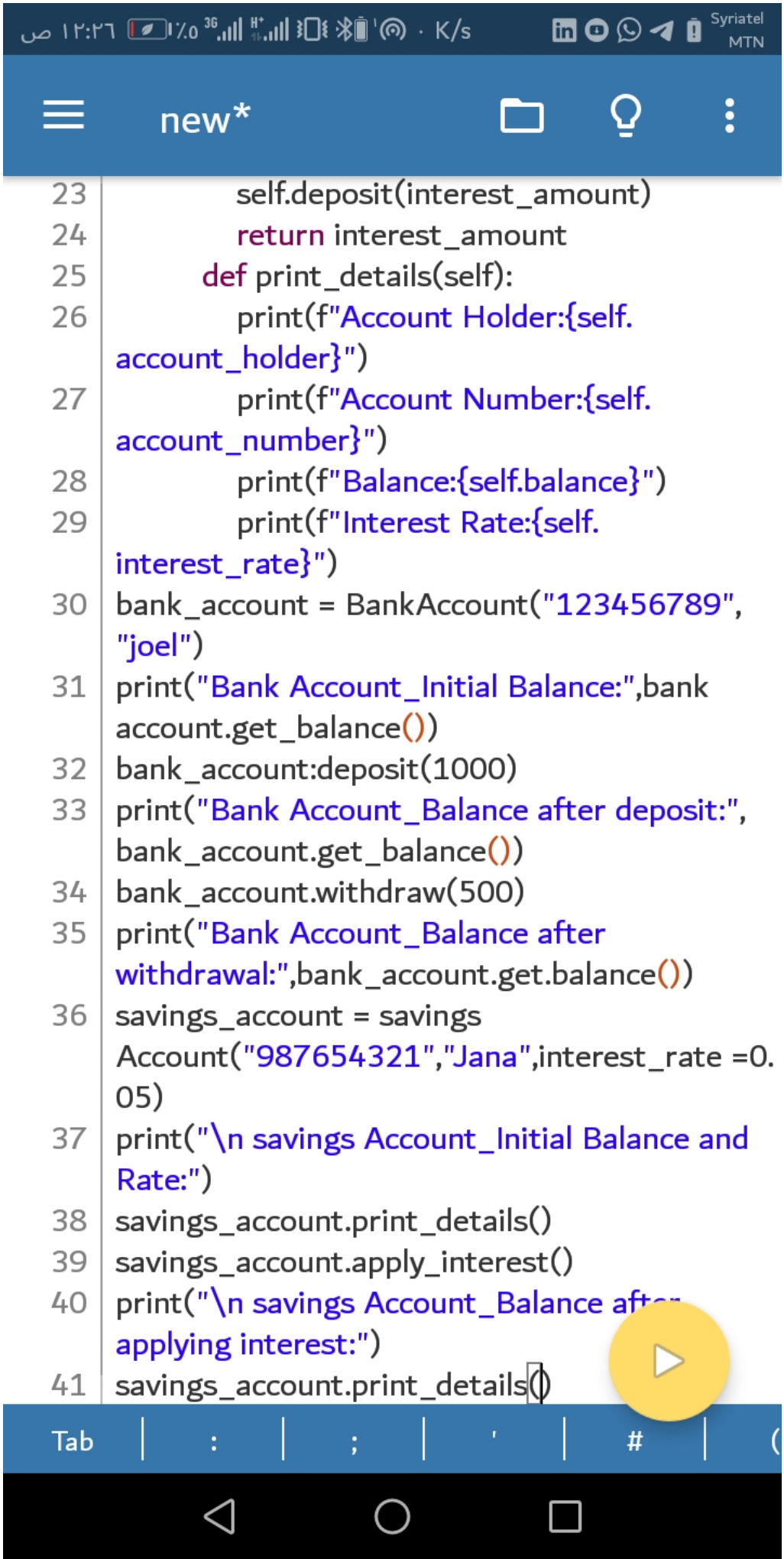

١١:٤٥ م ٢١٪ ١٤٢ B/s Syriatel MTN
1 import csv
2 def load_quiz_data(file_path):
3 quiz_data = []
4 with open(file_path,'r') as file:
5 csv_reader = csv.reader(file)
6 for row in csv_reader:
7 question,answer = row
8 quiz_data.append((question,answer))
9 return quiz_data
10 def take_quiz (quiz_data):
11 score =0
12 for question,answer in quiz_data:
13 user_answer =input(question +" ")
14 if user_answer.lower() == answer.
lower():
15 score +=1
16 return score
17 def save_user_result (user_name,score):
18 with open('user_results.csv','a',newline ='')
as file:
19 csv_writer = csv.writer(file)
20 csv_writer.writerow([user_name,score])
21 def main():
22 file_path ='quiz_data.csv'
23 quiz_data = load_quiz_data(file_path)
24 user_name = input("Enter your name:")
25 score =take_quiz(quiz_data)
26 print(f"your score: {score}")
27 save_user_result(user_name,scor
28 if_name_ == "_main_":
Tab : ; ' # (


new*
quiz_data = []
with open(file_path,'r') as file:
csv_reader = csv.reader(file)
for row in csv_reader:
question,answer = row
quiz_data.append((question,answer))
return quiz_data
def take_quiz (quiz_data):
score =0
for question,answer in quiz_data:
user_answer =input(question +" ")
if user_answer.lower() == answer.
lower():
score +=1
return score
def save_user_result (user_name,score):
with open('user_results.csv','a',newline ='')
as file:
csv_writer = csv.writer(file)
csv_writer.writerow([user_name,score])
def main():
file_path ='quiz_data.csv'
quiz_data = load_quiz_data(file_path)
user_name = input("Enter your name:")
score =take_quiz(quiz_data)
print(f"your score: {score}")
save_user_result(user_name,score)
if_name_ == "_main_":
main()
Tab
:
;
'
#


١١:٤٢ م
١٧٥ B/s
Syriatel
MTN
TAB
[Program finished]

:Question 4


new*
class BankAccount:
def_init_(self,account_number,
account_holder,balance = 0.0):
self.account_number =
account_number
self.account_holder = account_holder
self.balance = balance
def deposit(self,amount):
self.balance +=amount
return self.balance
def withdraw(self,amount):
if amount > self.balance:
print("Insufficient funds")
else:
self.balance = amount
return self.balance
def get_balance(self):
return self.balance
class savings Account(Bank Account):
def_init_(self,account_number,
account_holder,balance = 0.0,interest_rate = 0.
0)
super()._init_(account_number,
account_holder,balance)
self.interest_rate = interest_rate
def apply_interest(self):
interest_amount = self.balance*self.
interest_rate
self.deposit(interest_amou
return interest_amount
Tab
:
;
'
#




```
            self.deposit(interest_amount)
            return interest_amount
        def print_details(self):
            print(f"Account Holder:{self.
account_holder}")
            print(f"Account Number:{self.
account_number}")
            print(f"Balance:{self.balance}")
            print(f"Interest Rate:{self.
interest_rate}")
bank_account = BankAccount("123456789",
"joel")
print("Bank Account_Initial Balance:",bank
account.get_balance())
bank_account:deposit(1000)
print("Bank Account_Balance after deposit:",
bank_account.get_balance())
bank_account.withdraw(500)
print("Bank Account_Balance after
withdrawal:",bank_account.get.balance())
savings_account = savings
Account("987654321","Jana",interest_rate =0.
05)
print("\n savings Account_Initial Balance and
Rate:")
savings_account.print_details()
savings_account.apply_interest()
print("\n savings Account_Balance after
applying interest:")
savings_account.print_details()
```