[Python] Important Algorithms

Factorial function

```
def factorial(n):   #Task 1: Factorial function
    if n==0:    #Case 1: 0!
        return 1

    else:   #Case 2: <positive_integer>!
        return n*factorial(n-1) #multiply with factorial of n-1
```

Recursive arithmetic sum

```
def arithmetic_sum_2(term1,com_dif,n):   #Task 2(alternative): Arithmetic Sum Function — Recursive method
    if n==1:    #Case 1: last term to add(=the first term of the sequence)
        return term1
    else:   #Case 2: add <positive integer>th term
        return arithmetic_sum_2(term1,com_dif,(n-1))+(n-1)*com_dif+term1
```

Iterative arithmetic sum

```
def arithmetic_sum_1(term1,com_dif,n):   #Task 2: Arithmetic Sum function — Formula method
    return (n/2)*(2*term1+(n-1)*com_dif)
```

Recursive geometric sum

```
def geometric_sum_2(term1,com_rat,n):   #Task 3: Geometric Sum function — Recursive method
    if n==1:    #Case 1: last term to add(=the first term of the sequence)
        return float(term1)
    else:   #Case 2: add <positive integer>th term
        return float(geometric_sum_2(term1,com_rat,(n-1))+term1*(com_rat**(n-1)))  #perform
```

```
nth term addition
```

Iterative geometric sum

```
def geometric_sum_1(term1,com_rat,n):   #Task 3: Geometric Sum function – Formula method
    return (term1-(com_rat**n))/(1-com_rat)
```

Fibonacci function

```
def fibonacci(term1,term2,n):   #Task 4: Fibonacci function
    if n == 0:
        return term1
    elif n == 1:
        return term2
    else:
        return fibonacci(term1,term2,(n-1))+fibonacci(term1,term2,(n-2))
```

Compound Interest

```
def compound_interest(initial,inc,n):
    return initial*((1+(inc/100))**n)
```

PIN, deposit & withdraw

```
correct_pin=1111
enter_pin=0
attempt=0
balance=100.00
option=0
save=0.00
password=True
while attempt<3 and password:
    while True:
        try:
```

```
            enter_pin=str(input("Enter PIN (4 digits)"))
            if len(str(enter_pin))==4 and int(enter_pin)>=0 and int(enter_pin)<=9999:
                break
            else:
                print("invalid format")
        except ValueError:
            print("invalid format")
    if int(enter_pin)==int(correct_pin):
        password=False
    else:
        attempt+=1
        print("wrong")

def deposit():
    global balance
    while True:
        try:
            save=float(input("How much you save"))
            if save>0:
                break
            else:
                print("can't save that amount")
        except ValueError:
            print("invalid format")
    balance+=save

def withdraw():
    global balance
    while True:
        try:
            take=float(input("How much you withdraw"))
            if take>0 and take<=balance:
                break
            else:
                print("can't withdraw that amount")
        except ValueError:
            print("invalid format")
    balance-=take

def change_pin():
```

```
    global correct_pin
    while True:
        try:
            new=str(input("Enter new PIN(4digits)"))
            if len(str(new))==4 and int(new)>=0 and int(new)<=9999 and new!=correct_pin:
                break
            elif new==correct_pin:
                print("there is no change")
            else:
                print("invalid password")
        except ValueError:
            print("invalid format")
    correct_pin=new
    print("Change successful")

if not(password):
    Out=False
    while not(Out):
        print("MENU")
        print(f"Balance: {balance}")
        print("Deposit[1]")
        print("Withdraw[2]")
        print("Change PIN[3]")
        print("Exit[4]")
        while True:
            try:
                option=int(input("Enter Option(number key)"))
                if option>=1 and option<=4:
                    break
                else:
                    print("no such option")
            except ValueError:
                print("invalid format")
        if option==1:
            deposit()
        elif option==2:
            withdraw()
        elif option==3:
            change_pin()
        else:
```

```
            Out=True

else:
    print("Max attempt reached")

if Out==True:
    print("End.")
```

Revision #4
Created 2026-03-18 23:55:49 UTC by Samuel Lee
Updated 2026-03-19 00:34:01 UTC by Samuel Lee