

```
python
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n *
factorial(n-1)
```

You can use this function by passing a number as an argument, for example:

```
python
result = factorial(5)
print(result)
```

This will output `120`, which is the factorial of 5.

python

```
# Get the year from the user  
year = int(input("Enter a year:  
"))
```

```
# Check if it's a leap year  
if (year % 4 == 0 and year %  
100 != 0) or (year % 400 == 0):  
    print(f"{year} is a leap  
year.")  
else:  
    print(f"{year} is not a  
leap year.")
```



```
python
class BankAccount:
    def __init__(self, account_holder,
balance=0):
        self.account_holder = account_holder
        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited ${amount}. New
balance: ${self.balance}")
        else:
            print("Invalid deposit amount.")

    def withdraw(self, amount):
        if amount > 0 and amount <=
self.balance:
            self.balance -= amount
            print(f"Withdrew ${amount}. New
balance: ${self.balance}")
        else:
            print("Invalid withdrawal amount or
insufficient funds.")

    def check_balance(self):
        print(f"Account balance for
{self.account_holder}: ${self.balance}")
```



```
python
def
linear_search_product(products,
target_product):
    for index, product in
enumerate(products):
        if product ==
target_product:
            return index
    return -1 # Return -1 if
the target product is not found

# Example usage:
products_list = ["Laptop",
"Phone", "Tablet", "Camera",
"Headphones"]

target_product = input("Enter
the product you're searching
for: ")

index = linear_search_
product(products_list,
target_product)

if index != -1:
    print(f"{target_product}
found at index {index}.")
else:
    print(f"{target_product}
not found in the list.")
```

```

class Player:
    def __init__(self, name,
score=0):
        self.name = name
        self.score = score

    def increase_score(self,
points):
        if points > 0:
            self.score +=
points
            print(f"{self.name}
earned {points} points. Total
score: {self.score}")
        else:
            print("Invalid
points value. Score not
updated.")

    def
display_player_info(self):
        print(f"Player:
{self.name}, Score:
{self.score}")

```

```

# Example usage:
# Create players and manipulate
scores
player1 = Player("Alice")
player2 = Player("Bob")

```

```

player1.increase_score(10)
player2.increase_score(5)

```

```

player1.display player info()

```



```

def __init__(self, name,
student_id, courses=None):
    self.name = name
    self.student_id =
student_id
    self.courses = courses
    if courses is not None else []

    def enroll_course(self,
course):
        if course not in
self.courses:
            self.courses.append(course)
            print(f"{self.name}
enrolled in {course}.")
        else:
            print(f"{self.name}
is already enrolled in
{course}.")

    def
display_student_info(self):
        print(f"Student:
{self.name}, ID:
{self.student_id}")
        if self.courses:
            print("Courses
enrolled: * * *
".join(self.courses))
        else:
            print("No courses
enrolled yet.")

```