EXERCISES

FIND THE DISTANCE!

- ✓ Given four real numbers representing Cartesian coordinates: (x1,y1),(x2,y2) write a function distance(x1, y1, x2, y2) to compute the distance between the points (x1,y1) and (x2,y2).
- ✓ Insert as input four real numbers and print the resulting distance calculated by the function.
- ✓ Use the functions math.sqrt() and math.pow() from the math library.

Hint:

Distance between the two points formula

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

SOLUTION

```
import math
def dist(x1,y1,x2,y2):
    d=math.sqrt(math.pow(x2-x1,2)+math.pow(y2-y1,2))
    print("The distance is ",d)
```

FIBONACCI RECURSION

Write a function that gets an integer as input and returns the Fibonacci number with order equal to that integer.

Hint: The Fibonacci sequence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, …

SOLUTION

We use a recursion function here:

```
def fib(n) :
    if n == 1 or n == 2:
        return 1
    return fib(n-1) + fib(n-2)
```

N! FACTORIAL

Write a function that gets a natural number and returns the factorial of that input number. The function must print the following message:

“Factorial can not be computed for negative numbers”

when the input number is negative.

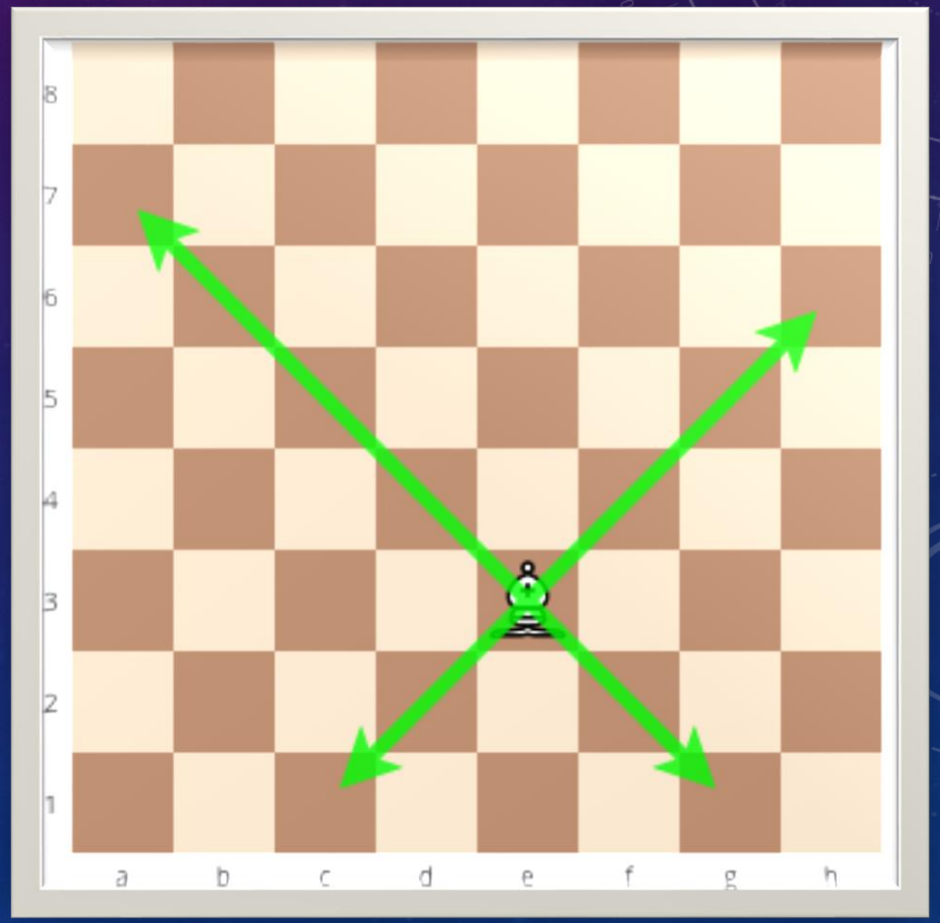
*Hint: n factorial is computed by: n!=1*2*3*…*n*

SOLUTION

```
def recur_factorial(n):
    if n < 0 :
        print("Factorial doesn’t exist for negative numbers")
    elif n == 0 or n == 1:
        return 1
    else:
        return n * recur_factorial(n-1)
```

BISHOP MOVES

- ✓ Write a function that receives as input four numbers from 1 to 8, specifying the column and row numbers of the starting and ending square.
- ✓ Print Yes if a Bishop can go from the first square to the second in one move, and otherwise No.
- ✓ Use the math.fabs() function from the math library.



HINT!

- In chess, the bishop moves diagonally any number of squares.
- In order to be able to move from (x1,y1) to (x2,y2) the following condition must hold:

```
math.fabs(x2-x1) == math.fabs(y2-y1)
```

SOLUTION

```
import math
def bis(x1,y1,x2,y2):
    if math.fabs(x2-x1)==math.fabs(y2-y1):
        print("Yes")
    else:
        print("No")
```

ROULETTE PAYOUTS

For this game a roulette has 37 spaces on it (18 black, 18 red, 1 green).

The green space is numbered 0.

Red spaces :

1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30 32, 34,36.

The remaining integers between 1 and 36 are black spaces.

➢Many different bets can be placed in roulette:

- Single number (1 to 36 or 0)
- Red versus Black
- Odd versus Even

(Note that 0 does not pay out for even)

✓ Change the code from the previous program,that simulated the spin of a roulette wheel, so as each type of bet to be a function.

✓ The goal is to use as many functions as you can in order to make the code more readable and reusable.

PREVIOUS PROGRAM

```
import random
money = 100
print("Welcome to the BIG BET!")
print("Press Enter to play!")

while( input() != ('s' or 'S')):
    print("Choose bet:\n Press 1 for number bet \n Press 2 for red-black bet \n Press 3 for an odd-even bet \n")
    choice = input("Press a button:")
    if choice == '1':
        gamble = int(input("\nGamble in number:"))
        temp = random.randint(0,36)
        print("\nThe spin resulted in ... %d" %temp)
        if gamble == temp:
            outcome = True
        else:
            outcome = False

    elif choice == '2':
        gamble = int(input("Select color:\n Press 1 for red \n Press 2 for black"))
        temp = random.randint(1,2)
        if temp == 1:
            color = 'red'
        else:
            color = 'black'
        print("\nThe spin resulted in ... " + color)
        if gamble == temp:
            outcome = True
        else:
            outcome = False

    elif choice == '3':
        num = int(input("Press 1 for odd and 2 for even.Select: "))
        if((temp % 2 ==0 and num == 2)or(temp % 2 != 0 and num == 1)):
            outcome = True
        else:
            outcome = False

    if outcome:
        print("You won 100 Euro!!!")
        money += 100
        print("Money remaining: %d" %money)
    else:
        print("You lost...Pay 10 Euro")
        money -= 10
        print("Money remaining: %d" %money)
    print("\n Press Enter to continue and s to exit")
```

SOLUTION

```
import random

def num_bet():
    gamble = int(input("\n Gamble in number:"))
    temp = random.randint(0,36)
    print("\n The spin resulted in... %d" %temp)
    if gamble == temp:
        outcome = True
    else:
        outcome = False
    return outcome

def col_bet():
    gamble = int(input("Select color:\n Press 1 for red \n Press 2 for black"))
    temp = random.randint(1,2)
    if temp == 1:
        color = 'red'
    else:
        color = 'black'
    print("\n The spin resulted in ..." + color)
    if gamble == temp:
        outcome = True
    else:
        outcome = False
    return outcome
```

```
def OddVsEven():
    num = int(input("Press 1 for odd and 2 for even.Select: "))
    temp = random.randint(0,36)
    if (temp % 2 == 0 and num == 2) or (temp % 2 != 0 and num == 1):
        outcome = True
    else:
        outcome = True
    return outcome

def outcome(x):
    money = 100
    if x:
        print("You won 100 Euro!!!")
        money += 100
        print("Money remaining: %d" %money)
    else:
        print("You lost...Pay 10 Euro")
        money -= 10
        print("Money remaining: %d" %money)

print("Welcome to the BIG BET!")
print("Press Enter to play!")

while(input() != ('s' or 'S')):
    print("Choose bet:\n Press 1 for a number bet \n Press 2 for a red-black")
    choice = input("Press a button:")
    if choice == '1':
        outcome(num_bet())
    elif choice == '2':
        outcome(col_bet())
    elif choice == '3':
        outcome(OddVsEvent())

    print("\nPress Enter to continue and s to exit the game")
```

BIBLIOGRAPHY

- Python Tutorial Release 3.6.4(Guido van Rossum and the Python development team) from www.python.org