Pythonprojekt för Borås högskoleprov

Bin Han

2025-07-17

Table of contents

1 Project Description 1

2 Assignments 1

2.1 Uppgift 1 . . . 1

2.2 Uppgift 2 . . . 2

2.3 Uppgift 3 . . . 3

2.4 Uppgift 4 . . . 4

2.5 Uppgift 5 . . . 5

2.6 Uppgift 6 . . . 5

2.7 Uppgift 7 . . . 6

2.8 Uppgift 8 . . . 7

2.9 Uppgift 9 . . . 8

1 Project Description

This assignment series comprises nine structured programming tasks designed to reinforce fundamental and intermediate Python skills. The problems span key topics such as control flow, user input, custom functions, numerical logic, and data structure manipulation, culminating in a hands-on introduction to data visualization using matplotlib. Students are required to implement constraint-based solutions, such as performing binary conversion without built-in functions and visualizing sorted versus unsorted data, emphasizing accuracy, modularity, and computational thinking across real-world scenarios.

2 Assignments

2.1 Uppgift 1

Responsible teacher: Markus

Write a Python program that receives one or more sentences from the user and, using a self-defined function, counts the number of occurrences of a specific word and returns these.

Example: If the user inputs the following string: “Python is a powerful and useful programming language. Python is used all over the world.” and the specific word is: “python”, then the program should count the number of occurrences of the word “python” and output the following result:

Number of occurrences of “python”: 2

Requirements:

The user should be able to input one or more sentences as a source. The user should be able to input a word to be used for matching. The program should not differentiate between uppercase and lowercase letters (case-insensitive). The program should count the number of occurrences of the specific search word in the text/source. No modules may be imported, in such cases you will receive 0 points for the assignment.

**When I render a .qmd (Quarto markdown) file, the code chunks usually run non-interactively, so input() calls won’t work because they require interactive user input. So I dicide to Replace input() with predefined variables in the .qmd”

```python
def word_counter(Source, word):
    occurrences=0
    for item in Source.split():
        if item.casefold()== word.casefold():
            occurrences+=1
    print(f"Number of occurrences is {occurrences}")
A="Python is a powerful and useful programming language. Python is used all over the world
B="python"
word_counter(A, B)
```

```
Number of occurrences is 2
```

2.2 Uppgift 2

Responsible teacher: Markus

Write a Python program that, using a self-defined function, calculates and returns the highest of three numbers entered by the user. Example: If the user enters the following numbers: 7, 15, 4, the program should calculate the highest of the numbers, which is 15, and output the following result: The highest number is: 15 Requirements: The user should be able to enter three numbers. The function should take three arguments and return the highest of the three numbers. Inputting numbers and printing to the screen may not take place in the above function. No modules may be imported, in such cases you will receive 0 points for the assignment.

```
def highest_finder(a, b, c):
    if a>=b and a>=c:
        return a
    elif b>=a and b>=c:
        return b
    else:
        return c
A=7
B=15
C=4
H=highest_finder(A, B, C)
print(f"The highest number is {H}")
```

```
The highest number is 15
```

2.3 Uppgift 3

Responsible teacher: Markus Create a Python program that handles simple temperature conversion. The program should contain two custom functions: one to convert Celsius to Fahrenheit and one to convert Fahrenheit to Celsius. The user should be able to choose which type of conversion to perform and enter a temperature. The program should then, by using the defined functions, convert the temperature and print the result on the screen. Requirements: Create a function called celsius_to_fahrenheit(celsius) that receives a temperature in Celsius and returns the corresponding temperature in Fahrenheit using the formula: F = C * 9/5 + 32. Create a function called fahrenheit_to_celsius(fahrenheit) that receives a temperature in Fahrenheit and returns the corresponding temperature in Celsius using the formula: C = (F - 32) * 5/9. Implement a menu where the user can choose which conversion they want to perform. The user should be able to enter a temperature. The program should use the defined functions to perform the conversion. Print the result of the conversion. Tip: Keep in mind that decimals must be able to be used, so which variable is appropriate?

```
def celsius_to_fahrenheit(celsius):
    return float(celsius)*9/5+32
def fahrenheit_to_celsius(fahrenheit):
    return (float(fahrenheit)-32)*5/9

c1=25
f1=celsius_to_fahrenheit(c1)
print(f"{c1} motsvarar {f1:.1f} Fahrenheit.")

f2=68
c2=fahrenheit_to_celsius(f2)
print(f"{f2} motsvarar {c2:.1f} Celsius.")
```

```
25 motsvarar 77.0 Fahrenheit.
68 motsvarar 20.0 Celsius.
```

2.4 Uppgift 4

Convert integers from decimal form to binary form

Responsible teacher: Peter

Computers use the binary number system to store information. A binary number is made up of a number of bits where each bit (from the English binary digit) has two possible values, either '0' or '1'. A bit is the foundation of the binary number system, which has a base of 2 (compared to our 'normal' number system, which has a base of 10). In order to make it practically possible to use the binary number system in a computer, a number is always represented by a certain number of bits, for example, a sequence of 8 bits is called a byte. However, to convert a number in decimal form to its binary representation, you can use the following method: Step 1: Use integer division and divide the integer by 2. Write down the integer quotient and the remainder for this division. Step 2: Use integer division and divide the previous integer quotient by 2. Write down the integer quotient and the remainder for this division. Step 3: Continue this sequence (step 2) until you get the quotient 0. Step 4: You can write the binary number as all the remainders in REVERSE order (read these from bottom to top in the example below). Example with a conversion of the integer 35 to its binary representation Integer Integer division Integer quotient Remainder Note 35 35 / 2 17 1 2 x 17 + 1 = 35 17 17 / 2 8 1 2 x 8 + 1 = 17 8 8 / 2 4 0 2 x 4 + 0 = 8 4 4 / 2 2 0 2 x 2 + 0 = 4 2 2 / 2 1 0 2 x 1 + 0 = 2 1 1 / 2 0 1 2 x 0 + 1 = 1 The integer 35 is therefore 100011 in binary form! Write a program where you use this method to convert integers, in decimal form, to their binary representation. No modules may be imported, including the function bin(). In such cases you will receive 0 points for the task.

```python
def DecimalConvertBinary(n):
    n1=n
    s=""
    while n1!=0:
        remainder=n1%2
        n1=n1//2
        s=s+str(remainder)
    s=s[::-1]
    print(F"The integers {n} is therefor {s} in binary form!")
DecimalConvertBinary(35)
```

```
The integers 35 is therefor 100011 in binary form!
```

2.5 Uppgift 5

Calculate number of tiles

Responsible teacher: Peter You will help the company Kakelplattan AB to write a program that calculates how many packages of tiles, cartons of tile fix, and bags of tile grout that the craftsman needs to bring to each customer. The program will ask what type of work it is and how many square meters it is. The program can handle three different types of work, Bathroom (B), Hall (H) and Kitchen (K). A package of tiles contains 32 tiles, a carton of tile fix weighs 5 kg and a bag of tile grout weighs 3 kg. A bathroom needs per square meter Tile tiles: 15 pieces Tile fix: 0.7 kg Tile joint: 0.4 kg A hall needs per square meter Tile tiles: 8 Tile fix: 0.3 kg Tile joint: 0.25 kg A kitchen needs per square meter Tile tiles: 12 pieces Tile fix: 0.2 kg Tile joint: 0.15 kg Note that the calculations must be rounded up because the craftsman cannot bring half a package to the customer. For example, if the answer is 6.1 packages of tiles, the program should round up to 7 packages of tiles so that the craftsman does not bring too little material No modules other than the math module may be imported into your program. If any additional modules are imported, you will receive 0 points for the task.

```python
import math
def PackageCompute(A, Num):# A ange type av arbete. Num ange antal kvadratmeter
    if A=="B":
       plattor=math.ceil(Num*15/32)
       fix=math.ceil(Num*0.7/5)
       fog=math.ceil(Num*0.4/3)
    elif A=="H":
       plattor=math.ceil(Num*8/32)
       fix=math.ceil(Num*0.3/5)
       fog=math.ceil(Num*0.25/3)
    else:
       plattor=math.ceil(Num*12/32)
       fix=math.ceil(Num*0.2/5)
       fog=math.ceil(Num*0.15/3)
    print(f"Ta med {plattor} paket Kakelplattor, {fix} kartong(er) Kakelfix samt {fog} säc

PackageCompute("K", 22)
```

```
Ta med 9 paket Kakelplattor, 1 kartong(er) Kakelfix samt 2 säck(ar) Kakelfog.
```

2.6 Uppgift 6

Rock, scissors or bag

Responsible teacher: Peter

Write a program that you can use to play “Rock, Scissors or Bag” against the computer. You probably know that rock wins over scissors, scissors wins over bag and bag wins over rock. If both choose the same option, it is a draw. Suitable solution: 1. Let the player choose one of the options rock, scissors or bag. 2. The program randomly selects which option of rock, scissors or bag the computer chooses. 3. Print what the player and the computer have chosen and who won. 4. Ask the player if he wants to play again. 5. This is repeated until the player does not want to play anymore. No modules other than the random module may be imported. In such cases, you will receive 0 points for the task.

```
import random
def StenSaxPåse(A, B):
    Lis=["sten", "sax", "påse"]
    A=random.choice(Lis)
    B=random.choice(Lis)
    print(f"Du valde {A}, Datorn valde {B}.")
    if (A=="sten" and B=="påse") or (A=="påse" and B=="sax") or (A=="sax" and B=="sten"):
        print(f"{B} vinner över {A}! Du förlorade!")
    elif (B=="sten" and A=="påse") or (B=="påse" and A=="sax") or (B=="sax" and A=="sten")
        print(f"{A} vinner över {B}! Du vann!")
    else:
            print(f"oavgjort!")
StenSaxPåse(A, B)
StenSaxPåse(A, B)
StenSaxPåse(A, B)
```

```
Du valde sten, Datorn valde sax.
sten vinner över sax! Du vann!
Du valde sten, Datorn valde sax.
sten vinner över sax! Du vann!
Du valde sten, Datorn valde sten.
oavgjort!
```

2.7 Uppgift 7

Responsible teacher: Håkan

Write a self-defined function called triangles that takes three arguments, the sides of the triangle. The function should test whether the triangle is right-angled by using the Pythagorean theorem: If the triangle is right-angled, the area A is calculated as and the perimeter O as If the triangle is not right-angled, the area A is calculated by using Heron’s formula: where s is and the perimeter is calculated as above. The function prints the results and whether the triangle is right-angled or not. The results should be of the data type float with 2 decimal places of accuracy. Show that the function works by writing a main program that asks the user how many triangles to test. For each triangle, the length of the three sides is then requested and the function is called and prints the results according to the following printout:

```
import math
def trianglar(a, b, c):
    if a**2+b**2==c**2:
        print(f"Triangle är rätvinklig.")
        A=a*b/2
    else:
        print(f"Trianglen är inte rätvinklig.")
        s=(a+b+c)/2
        A=math.sqrt(s*(s-a)*(s-b)*(s-c))
    O=a+b+c
    print(f"Triangelns area A={A:.2f} cm^2 och omkrtsen O={O:.2f} cm.")
trianglar(3, 4, 6)
trianglar(3, 4, 5)
```

```
Trianglen är inte rätvinklig.
Triangelns area A=5.33 cm^2 och omkrtsen O=13.00 cm.
Triangle är rätvinklig.
Triangelns area A=6.00 cm^2 och omkrtsen O=12.00 cm.
```

2.8 Uppgift 8

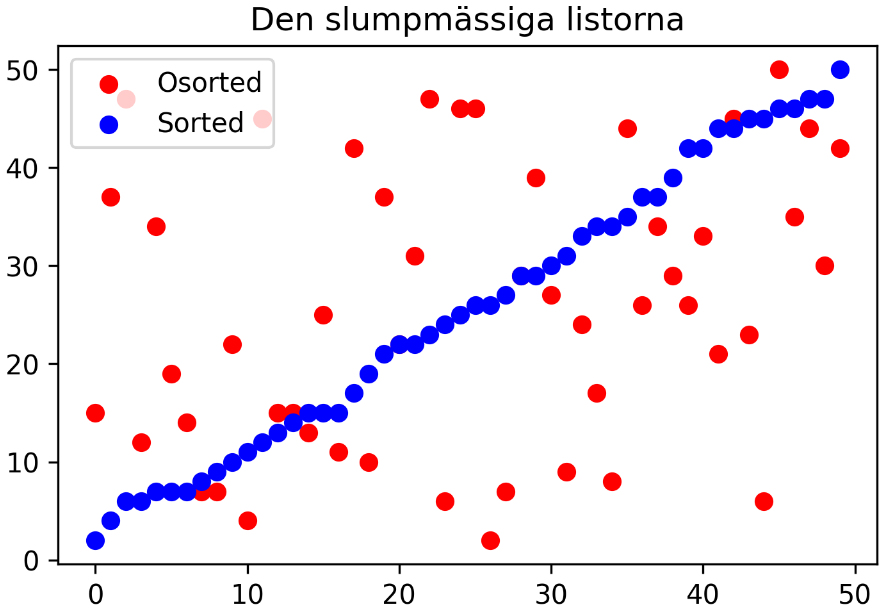
Responsible teacher: Håkan

Create a program that calls the module random and generates 50 random numbers (integers) between 1 and 50 and saves the random numbers in a list. The list should then be sorted into a new list so that the original list remains, after which both lists are printed on the screen. Then use the module matplotlib to draw a scatter plot that shows the contents of the two lists according to the following diagram:

```
import random
import matplotlib.pyplot as plt

lis=[]
for i in range(50):
    a=random.randint(1,50)
    lis.append(a)
print(f"Den slumpmässiga listan: {lis}")

lis1=sorted(lis)
print(f"Den sorted listan: {lis1}")

x=list(range(50))
```

```
plt.scatter(x, lis, color="red", label="Osorted")
plt.scatter(x, lis1, color="blue", label="Sorted")
plt.title("Den slumpmässiga listorna")
plt.xlable=("position")
plt.ylable=("Värde")
plt.legend(loc="upper left")
plt.show()
```

```
Den slumpmässiga listan: [15, 37, 47, 12, 34, 19, 14, 7, 7, 22, 4, 45, 15, 15, 13, 25, 11,
Den sorted listan: [2, 4, 6, 6, 7, 7, 7, 8, 9, 10, 11, 12, 13, 14, 15, 15, 15, 17, 19, 21,
```



2.9 Uppgift 9

Responsible teacher: Håkan

Study the program sequence below that prints all prime numbers between 1 and the number lastNumber. Unfortunately, there are three errors in the code that make it not work correctly. Correct the errors. Write a comment for each change you make, uncommented changes do not give points. No lines may be added or deleted. Definition of prime number: A prime number is a non-negative integer that can only be divided by the number one (1) or by itself to make an even number.

```
def Primefind(sistaTalet):
    raknare, flagga = 3, "true"
    primes= [1, 2]
    while (raknare <= sistaTalet):# raknare loop till sistaTalet
        for i in range (2, raknare):
            if ((raknare % i) == 0):#the remainder is 0
                flagga = 'false'# non prime number
                break
        if (flagga == 'true'):
            primes.append(raknare)
        flagga = 'true'
        raknare += 1
    print(f"Primtal upp till {sistaTalet}: {', '.join(map(str, primes))}")#convert a list
Primefind(30)

Primefind(40)

Primefind(50)

Primefind(60)
```

```
Primtal upp till 30: 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29
Primtal upp till 40: 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37
Primtal upp till 50: 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47
Primtal upp till 60: 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59
```