

Link para o repositório no github: <https://github.com/luiz-lvj/ces22-listas>

Link para a pasta da lista 1 no repositório: <https://github.com/luiz-lvj/ces22-listas/tree/master/lista1>

### Questão 1)

Para desenhar os quadrados como pedido, foi criada a função principal abaixo e as funções auxiliares *draw\_square* e *move\_next\_position*:

```
def draw_picture_squares(initial_size, increment, number_squares):  
    """  
    Creates the picture with multiple squares, based on the number of squares, the side of the smallest  
    (and innermost) and the increment to the squares in the outside  
    """  
    screen = turtle.Screen()  
    agent = turtle.Turtle()  
    agent.pencolor("pink")  
    agent.pensize(3)  
    square_size = initial_size  
    for i in range(number_squares-1):  
        draw_square(square_size, agent)  
        move_next_position(increment/2, agent)  
        square_size += increment  
    draw_square(square_size, agent)  
    screen.mainloop()
```

A saída do programa é:



### Questão 2)

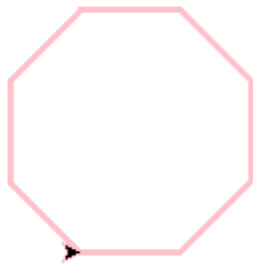
Foi criada a função *draw\_poly* para desenhar um polígono regular utilizando a biblioteca *turtle*:

```
import turtle

def draw_poly(tess, n, sz):
    """
    Draws a regular polygon, with number of sides equal to n and
    size of each side equal to sz. The parameter tess is an instance of the Turtle class
    """
    screen = turtle.Screen()
    angle = 360/n
    for i in range(n):
        tess.forward(sz)
        tess.left(angle)
    screen.mainloop()

if __name__ == "__main__":
    agent = turtle.Turtle()
    agent.pencolor("pink")
    agent.pensize(3)
    draw_poly(agent, 8, 50)
```

A saída do programa (um octógono de lado 50) é:



### Questão 3)

Nesse programa, foi criada uma função para retornar a soma de todos os inteiros de 1 a n, incluindo n. Além disso, todos os erros foram tratados, de modo que o programa para a execução em caso de input incorreto:

```

import sys
def sum_to(n):
    """
    Returns the sum of all integer numbers up to and including n.
    If n is not an integer or n < 0, the return is 0.
    """
    if type(n) != int or n < 0:
        print("Entrada inválida.")
        sys.exit()
    return int(n*(n+1)/2)

if __name__ == "__main__":
    input_n = 0
    try:
        input_n = int(input("Insira o número de entrada: "))
    except:
        print("Entrada inválida.")
        sys.exit()
    ans = sum_to(input_n)
    print("Soma: ", ans)

```

A saída do programa, para uma entrada igual a 10 é:

```

Insira o número de entrada: 10
Soma: 55

```

#### Questão 4)

Nessa questão foi criada uma função para testar se o número da entrada é primo ou não e, se a entrada for inválida, sair de execução. Para fazer um algoritmo melhor, foi usado o fato de que só é necessário testar os divisores de um inteiro até o piso da raiz desse número. O código é:

```

import math
import sys

def is_prime(n):
    """
    Returns True if the given integer n is prime and False otherwise
    """
    if n <= 1:
        return False
    lim_test = math.floor(math.sqrt(n))
    x = 2
    while x <= lim_test:
        if n % x == 0:
            return False
        x += 1
    return True

if __name__ == "__main__":
    input_n = input("Digite o número a ser testado: ")
    try:
        n = int(input_n)
    except:
        print("Entrada inválida.")
        sys.exit()
    print("É primo: ", is_prime(n))

```

A saída do programa para entradas 17 e 20, por exemplo, são:

```

Digite o número a ser testado: 17
É primo: True

```

```

Digite o número a ser testado: 20
É primo: False

```

#### Questão 5)

Nessa questão foi criado um algoritmo que testa se a string inputada é um palíndromo ou não. O código é:

```
def is_palindrome(input_str):  
    """  
    Returns True if the string input is a palindrome and False otherwise.  
    """  
    if len(input_str) == 0:  
        return True  
    if len(input_str) == 1:  
        return True  
    size = len(input_str)  
    if input_str[0] != input_str[size-1]:  
        return False  
    new_str = input_str[1:size-1]  
    return is_palindrome(new_str)  
  
if __name__ == "__main__":  
    input_str = input("Digite a string de entrada: ")  
    print("É palíndromo: ", is_palindrome(input_str))
```

A saída do programa para a string de entrada *arara* e *abcdefghi* são, respectivamente:

```
Digite a string de entrada: arara  
É palíndromo: True
```

```
Digite a string de entrada: abcdefghi  
É palíndromo: False
```