Curso:

**UPskill – Programação em Python**

Unidade de Formação:

**Programação**

Enunciados trabalhos

**['2023-11-24', '14:00', '17:30', 'SA']**

Entrega:

**2023-11-28 23:59**

Formato de entrega:

**PDF, baseado no formato existente no moodle (formato trabalho 1)**

Índice

[1 Suspeito de crime 2](#_Toc151575820)

[1.1 Teste do Programa 2](#_Toc151575821)

[2 Folha de recibo 2](#_Toc151575822)

[2.1 Teste do Programa 3](#_Toc151575823)

[2.1.1 Caso: 1 3](#_Toc151575824)

[2.1.2 Caso: 2 3](#_Toc151575825)

[2.1.3 Caso: 3 4](#_Toc151575826)

[2.1.4 Caso: 4 4](#_Toc151575827)

[2.1.5 Caso: 5 4](#_Toc151575828)

[2.1.6 Caso: 6 4](#_Toc151575829)

[2.1.7 Caso: 7 4](#_Toc151575830)

[2.1.8 Caso: 8 4](#_Toc151575831)

[3 Python 4](#_Toc151575832)

[3.1 Replicar texto: 5](#_Toc151575833)

[3.2 Formatação saídas 5](#_Toc151575834)

[3.3 Funções de string: ljust, rjust, center and zfill 5](#_Toc151575835)

[3.4 Dicas para trabalho 6](#_Toc151575836)

# Suspeito de crime

Faça um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são:

1. Telefonou para a vítima?
2. Esteve no local do crime?
3. Mora perto da vítima?
4. Devia para a vítima?
5. Já trabalhou com a vítima?

O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita", entre 3 e 4 como "Cúmplice" e 5 como "Assassino". Caso contrário, ele será classificado como "Inocente". Faça todas as validações de entrada de dados (tipos e limites) e formação dos resultados.

## Teste do Programa

Teste o programa para todos os casos que achar necessário para garantir que está correto.

# Folha de recibo

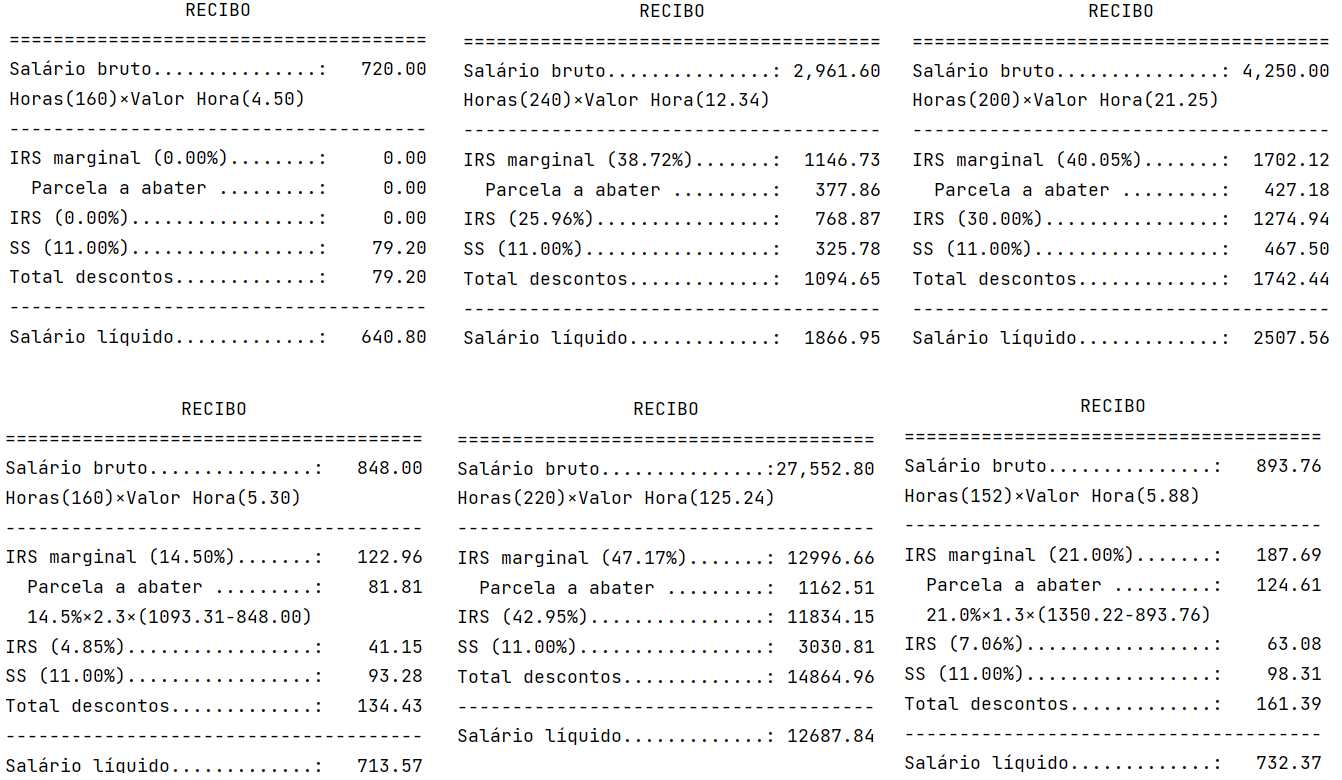
Faça um programa para o cálculo de uma folha de pagamento, sabendo que os descontos são o Imposto de Rendimentos para Pessoas singulares (IRS), que depende do salário bruto (conforme tabela abaixo) e 11% do Salário Bruto (SB) para a Segurança Social (SS).

O Salário Líquido (SI) corresponde ao SB menos os descontos.

O programa deverá pedir ao utilizador o valor da sua hora e a quantidade de horas trabalhadas no mês.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tabelas de IRS para o segundo semestre de 2023** | |  |
|  | **Trabalhadores** |  |  |
|  | **I — Trabalho dependente: Não casado ou casado dois titulares, sem dependentes** | | |
|  | **Remuneração** | **Taxa marginal** | **Parcela** |
|  | **mensal (€)** | **máxima** | **a abater (€)** |
| 1 | 762,00 | 0,00% | 0,00 |
| 2 | 886,57 | 14,50% | 14,5% x 2,3 x (1 093,31 - R) |
| 3 | 932,14 | 21,00% | 21,0% x 1,3 x (1 350,22 - R) |
| 4 | 999,14 | 21,00% | 114,14 |
| 5 | 1 106,93 | 26,50% | 169,09 |
| 6 | 1 600,36 | 28,50% | 191,23 |
| 7 | 1 961,36 | 35,00% | 295,26 |
| 8 | 2 529,05 | 37,00% | 334,48 |
| 9 | 3 694,46 | 38,72% | 377,86 |
| 10 | 5 469,90 | 40,05% | 427,18 |
| 11 | 6 420,55 | 42,72% | 573,22 |
| 12 | 20 064,21 | 44,95% | 716,08 |
| 13 | Superior a 20 064,21 | 47,17% | 1 162,51 |
|  | <https://www.montepio.org/ei/pessoal/impostos/tabelas-de-irs-saiba-quanto-vai-descontar/> | | |

Faça todas as validações de entrada de dados (tipos e limites) e formação dos resultados. Imprima no ecrã as informações, dispostas conforme nos exemplos abaixo (pode melhorar a apresentação do RECIBO se assim o entender).



## Teste do Programa

Teste o programa para os seguintes casos:

### Caso: 1

Horas......: 40

Valor Hora.: 10.00

### Caso: 2

Horas......: 160

Valor Hora.: 4.50

### Caso: 3

Horas......: 160

Valor Hora.: 5.30

### Caso: 4

Horas......: 152

Valor Hora.: 5.88

### Caso: 5

Horas......: 180

Valor Hora.: 6.59

### Caso: 6

Horas......: 240

Valor Hora.: 12.34

### Caso: 7

Horas......: 200

Valor Hora.: 21.25

### Caso: 8

Horas......: 220

Valor Hora.: 125.24

# Python

* Entrada de dados: [input](https://python-course.eu/python-tutorial/input.php)
* Conversão de dados: [int](https://www.w3schools.com/python/python_numbers.asp), [float](https://www.w3schools.com/python/python_numbers.asp)
* Formatação saídas: [print](https://python-course.eu/python-tutorial/formatted-output.php)
* Ciclos: [while](https://python-course.eu/python-tutorial/loops.php)
* Estrutura de decisão: [if, elif, else](https://python-course.eu/python-tutorial/conditional-statements.php)
* Erros e tratamento de exceções: [try … except](https://python-course.eu/python-tutorial/errors-and-exception-handling.php)

With the aid of exception handling, we can write robust code for reading an integer from input:

**while** True:

**try**:

n = **input**("Please enter an integer: ")

n = **int**(n)

**break**

**except** **ValueError**:

**print**("No valid integer! Please try again ...")

**print**("Great, you successfully entered an integer!")

## Replicar texto:

pontos = **'.'** \* 10  
print(x)  
print(pontos)  
  
*#12345678901234567890  
#..........*print(x)  
print(**f"{'':.<10}"**)  
*#12345678901234567890  
#..........*

## Formatação saídas

number\_string = 340020.8  
*# This portion is responsible for grouping the number*number\_commas\_only = **"{:,}"**.format(number\_string)  
print(number\_commas\_only)  
  
*# To ensure we have two decimal places*number\_two\_decimal = **"{:.2f}"**.format(number\_string)  
print(number\_two\_decimal)  
  
*# Both combined along with the currency symbol(in this case $)*currency\_string = **"{:,.2f}€"**.format(number\_string)  
print(currency\_string)  
  
*# 340,020.8  
# 340020.80  
# 340,020.80€*

<https://python-course.eu/python-tutorial/formatted-output.php>

## Funções de string: ljust, rjust, center and zfill

*# The string class contains further methods, which can be used for formatting purposes as well:  
# ljust, rjust, center and zfill.*print(**'\nS.center(width[, fillchar]) -> str'**)  
x = **'12345678901234567890'**s = **"Python"**print(x)  
print(s.center(10), **'.'**, sep=**''**)  
*#12345678901234567890  
# Python*y = s.center(10, **"\*"**)  
print (y)  
*# \*\*Python\*\**print(**'\nS.ljust(width[, fillchar]) -> str'**)  
s = **"UPSkills"**y = s.ljust(15)  
print(x)  
print(y, **'.'**, sep=**''**)  
  
y = s.ljust(15, **'\_'**)  
print(x)  
print(y, **'.'**, sep=**''**)  
  
print(**'\nS.rjust(width[, fillchar]) -> str'**)  
y = s.rjust(15)  
print(x)  
print(y, **'.'**, sep=**''**)  
  
y = s.rjust(15, **'\_'**)  
print(x)  
print(y, **'.'**, sep=**''**)  
  
print(**'\nS.zfill(width) -> str'**)  
s = **"4567"**y = s.zfill(15)  
print(x)  
print(y, **'.'**, sep=**''**)

S.ljust(width[, fillchar]) -> str  
12345678901234567890  
UPSkills .  
12345678901234567890  
UPSkills\_\_\_\_\_\_\_.  
  
S.rjust(width[, fillchar]) -> str  
12345678901234567890  
 UPSkills.  
12345678901234567890  
\_\_\_\_\_\_\_UPSkills.  
  
S.zfill(width) -> str  
12345678901234567890  
000000000004567.

## Dicas para trabalho

tirs = 12.25  
virs = 123.45  
sb = 1350.25  
ss = round(sb \* 0.11, 2)  
largura = 25  
tamanho = 10  
print(3 \* x)  
print(**f"Salário bruto"**.ljust(largura, **'.'**) + **":"**, **f"{**sb**:{**tamanho**},.2f}€"**)  
print(**f"IRS ({**tirs**:.2f}%)"**.ljust(largura, **'.'**) + **":"**, **f"{**virs**:{**tamanho**},.2f}€"**)  
print(**f"SS (11%)"**.ljust(largura, **'.'**) + **":"**, **f"{**ss**:{**tamanho**},.2f}€"**)  
*# Salário bruto............: 1,350.25€  
# IRS (12.25%).............: 123.45€  
# SS (11%).................: 148.53€*largura = 20  
tamanho = 8  
print(3 \* x)  
print(**f"Salário bruto"**.ljust(largura, **'.'**) + **":"**, **f"{**sb**:{**tamanho**},.2f}€"**)  
print(**f"IRS ({**tirs**:.2f}%)"**.ljust(largura, **'.'**) + **":"**, **f"{**virs**:{**tamanho**},.2f}€"**)  
print(**f"SS (11%)"**.ljust(largura, **'.'**) + **":"**, **f"{**ss**:{**tamanho**},.2f}€"**)  
  
*# 123456789012345678901234567890123456789012345678901234567890  
# Salário bruto.......: 1,350.25€  
# IRS (12.25%)........: 123.45€  
# SS (11%)............: 148.53€*