INF100 – Introdução à Programação I
Roteiro Prática 04 a 06 de abril de 2022

Como nesta semana não haverá aula prática na 5ª. feira, em função do recesso da Semana Santa, este roteiro da prática semanal está sendo liberado na 2ª. feira e deverá ser entregue até 4ª. feira, 23h59.

Para quem tiver segurança de programar usando os comandos de entrada e saída é possível fazer toda a tarefa com as informações da seção “Roteiro de Prática”.

Para aqueles que ainda não estão seguros de como os comandos de entrada e saída funcionam, sugerimos ler com atenção e executar TODOS os exemplos dados no Anexo II.

O Anexo I é de leitura obrigatória, para garantir que a implementação está usando corretamente as variáveis.

ATENÇÃO: nesta semana a presença na aula prática não é obrigatória! A presença será computada pela entrega do trabalho prático 2.

Roteiro de Prática

Nome do arquivo a ser entregue: **p02.py**
Obs.: Recomenda-se salvar o arquivo com certa frequência para não perder a digitação já feita em caso de uma falha na rede elétrica.

Faça o “download” do arquivo **p02.py**. Em seguida, entre o IDLE e abra o arquivo através do menu **File > Open...**.

Segue abaixo a figura com o trecho de código que você deverá ver na tela:

```
# Nome do aluno:
# Matrícula:
# Data:
# (breve comentário dizendo o que o programa faz)

nomeInst = 'Universidade Federal de Viçosa'
num_alunos = 16560      # alunos de graduação matriculados no campus Viçosa
rel_cand_vagas = 10.3   # relação candidatos / vagas total
area = 2353.94          # área física total em hectares (ha)
anoAtual = 2023         # ano corrente


print()    # este comando imprime uma linha em branco
```

Logo abaixo da última linha do programa, acrescente os comandos necessários para que o programa faça as seguintes tarefas:

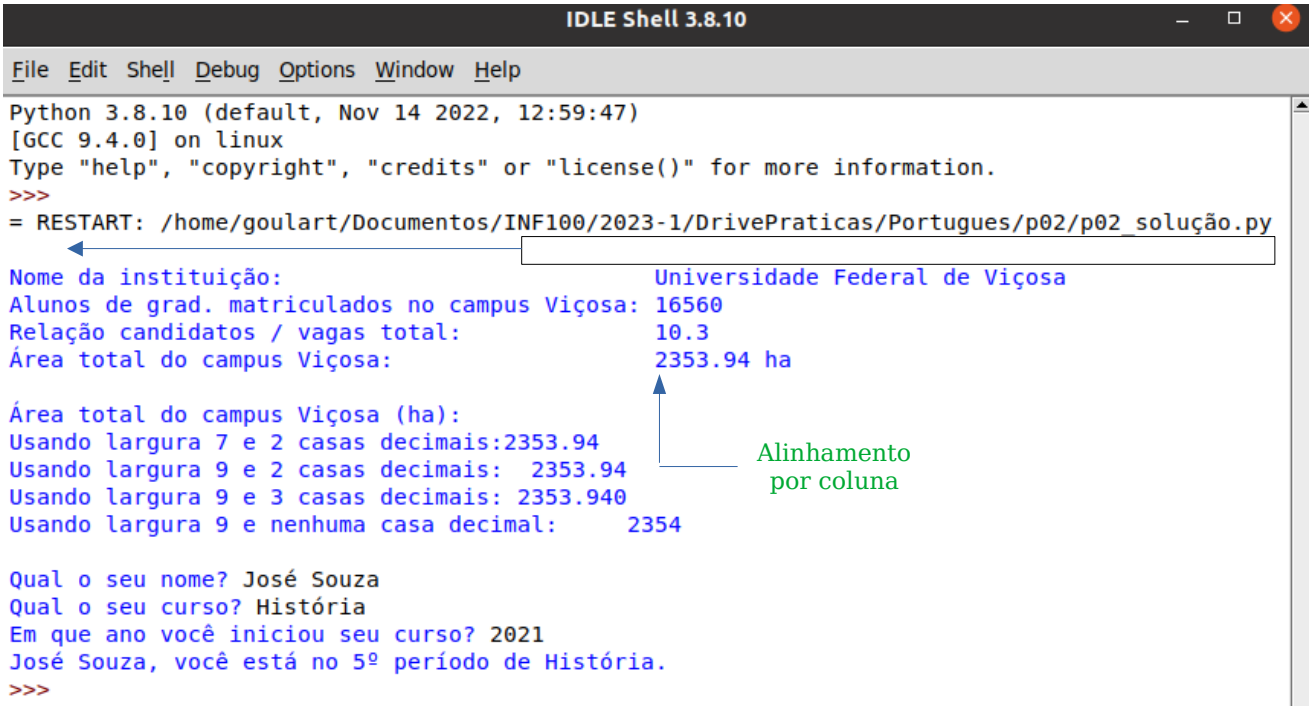
- imprimir as informações sobre a instituição, organizadas em dois blocos:
    - o primeiro bloco com o nome, número de alunos de graduação, a relação candidatos por vaga e a área do campus;
    - o segundo bloco exibindo a informação da área do campus em diferentes formatos.

- na sequência o programa deverá solicitar ao usuário 3 informações: nome, curso e ano de entrada.

- finalmente, o programa deverá calcular e informar o período que o usuário está cursando.

Dica: considere que cada ano tem 2 períodos e que estamos no primeiro período de 2023. Para o cálculo, utilize a variável ano_atual que armazena o valor do ano corrente (2023), e o ano digitado pelo usuário (que deverá ser armazenado em uma variável que você irá criar).

A figura a seguir mostra a saída gerada na janela do Shell para uma execução do programa, após ter sido completado corretamente. As informações escritas pelo programa aparecem na cor AZUL, enquanto que as entradas fornecidas pelo usuário aparecem na cor PRETA:



O seu programa deve gerar uma saída **exatamente** como a mostrada no exemplo acima.

OBS: As caixas de texto com os textos em cor verde "1 linha em branco, resultado do comando print()." e "Alinhamento por coluna" e as setas azuis NÃO fazem parte da saída gerada pelo programa.

Antes de enviar seu programa

Não esqueça de preencher o cabeçalho com seus dados e uma breve descrição do programa (pode usar a mesma descrição mostrada no quadro acima).

Confira se o arquivo a ser enviado está com o nome obrigatório: **p02.py**

Confira, também, se o arquivo p02.py contém a versão final do seu programa (verifique a data e hora que mostra a última vez em que o arquivo foi salvo).

Envie o arquivo do programa fonte (**p02.py**) através do sistema do LBI:

http:linux-server.lbi.ufv.br

O prazo de entrega encerra-se às 23h59 de quarta-feira (05/04).

Os anexos deste roteiro foram elaborados para auxiliar na obtenção de uma solução correta.

O Anexo I deve ser lido **por todos os alunos** para fazer a solução usando as variáveis ao invés de textos fixos.

O Anexo II deve ser lido por aqueles que ainda estão com dúvidas sobre a utilização dos comandos de entrada (input) e de saída (print). Informações mais completas sobre os comandos de entrada e saída estão no capítulo 4 do livro texto.

Anexo I - Comentários para fazer a coisa certa

Esta seção deve ser lida para ajudar a obter a saída CORRETA e a mais parecida possível com a saída mostrada na Figura acima.

a) Nesta prática **NÃO** será permitido colocar os dados diretamente dentro dos comandos **print**. Por exemplo, suponha que a variável alunos_pos armazenasse o número de estudantes de pós-graduação:

```
alunos_pos = 3152  # número de estudantes da pós-graduação
```

NÃO faça assim:

```
print('Alunos de pós-graduação da UFV: 3152')
```

Ao invés de usar o texto 3152 você deve usar a variável, importando o valor para dentro do texto:

```
print('Alunos de pós-graduação da UFV: %d' %alunos_pos)
```

Ou seja, você deverá usar as variáveis criadas no início do programa para escrever seus dados da tela.

Neste pequeno exemplo isso pode parecer perda de tempo. Mas suponha que este valor fosse usado em 10 comandos print e, alguém observasse que o valor correto é 3151 ao invés de 3152. Para quem usou o texto, será necessário alterar o valor nos 10 comandos prints. No segundo caso basta alterar o valor atribuído para a variável *alunos_pos*.

b) Repare também a linha em branco existente antes da 1ª linha escrita na tela.

c) Você pode consultar os slides da Aula 01 para ver outros exemplos do uso do comando **print** com formatação.

Anexo II – Exemplos de comandos de entrada e saída

Considere o seguinte programa bem pequeno escrito em Python:

```
peso = 75.57
print('Peso:', peso, 'kg')
```

Esse programa escreverá na tela o seguinte:

```
Peso: 75.57 kg
```

Note que o comando **print()** escreverá 3 informações: o texto 'Peso:'; em seguida o valor da variável **peso** definido anteriormente; e logo depois o outro texto 'kg'. Note também que o **print()**, por padrão, insere um espaço em branco entre cada uma das informações impressas.

Para que o texto 'kg' saia colado ao número contido na variável ***peso***, temos que informar ao **print()** para não usar nenhum separador entre os parâmetros. Mas nesse caso, devemos inserir um espaço em branco depois do texto 'Peso: ' mais especificamente, após o caractere '**:**'

```
peso = 75.57
print('Peso: ', peso, 'kg', sep='')
```

O resultado será o seguinte:

```
Peso: 75.57kg
```

Podemos obter o mesmo resultado usando um recurso muito interessante de formatação de números e textos disponível em Python. Com este tipo de utilização, o comando print pode passar a ter apenas um único texto a ser impresso (que estará sempre entre aspas) e ter uma ou mais informações importadas para "dentro" do texto, usando uma diretiva de importação e formatação. Veja este exemplo, que importa um número de ponto flutuante para dentro do texto:

```
peso = 75.57
print('Peso: %.2fkg' %peso )
```

Nesse caso, o '%.2f' indica que se deseja formatar um valor real, também chamado de "valor de ponto flutuante", ou simplesmente **float** (que é nome deste tipo de informação na linguagem Python), usando duas casas decimais. Nesse caso devemos usar o caractere **%**, após o final do texto, seguida do nome da variável que contém o valor a ser importado.

Caso a intenção seja escrever o valor contido na variável peso com UMA casa decimal, basta fazer:

```
peso = 75.57
print('Peso: %.1fkg' % peso )
```

e o resultado será:

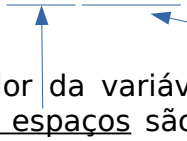
```
Peso: 75.6kg
```

Podemos também controlar a "largura" que o valor ocupar na tela, assim:

```
peso = 75.57
print('Peso:%10.3fkg' % peso )
```

Nesse caso, o resultado será:

```
Peso:    75.570kg
```



Como o valor da variável (75.570) ocupa 6 posições – o ponto decimal ocupa uma posição – 4 espaços são impressos na tela, para completar a largura de 10 espaços. Veja que nesse exemplo removemos o espaço em branco logo depois do ':', caso contrário teríamos 5 espaços em branco entre o ':' e o primeiro dígito do valor da variável *peso*.

Podemos também formatar valores inteiros, usando o formatador ‘%d’ (de “decimal”):

```
idade = 18
print('Tenho %d anos.' % idade )
```

Nesse caso, o resultado será:

```
Tenho 18 anos.
```

Assim como o formatador ‘%f’, com o ‘%d’ podemos controlar também a largura do (ou o espaço a ser ocupado pelo) valor impresso:

```
idade = 18
print('Tenho%4d anos.' % idade )
```

Nesse caso, o resultado será:

```
Tenho  18 anos.
```

Para preencher o lado esquerdo do número com zeros em vez de espaços, fazemos assim:

```
idade = 18
print('Tenho%04d anos.' % idade )
```

Nesse caso, o resultado será:

```
Tenho0018 anos.
```

Para formatar variáveis do tipo “texto” em vez de números, usamos o formatador “%s” (de *string*):

```
jantar = 'pizza'
print('Ontem jantei %s.' % jantar )
```

Isso escreverá na tela o seguinte:

```
Ontem jantei pizza.
```

Usando então esses formatadores e espaços em branco inseridos manualmente no texto entre aspas, podemos formatar e alinhar os valores da forma como quisermos.

Podemos também inserir vários números e textos na mesma linha escrita em tela. Só precisamos tomar o cuidado de colocar a lista de variáveis entre parênteses. Segue um exemplo:

```
idade = 18
rango = 'pizzas'
print('Tenho %d anos e ontem comi %02d %s.' % (idade, 3, rango) )
```

Isso escreverá na tela o seguinte:

```
Tenho 18 anos e ontem comi 03 pizzas.
```

Para permitir que o usuário do programa entre com um valor, devemos usar o comando input:

```
nome = input('Digite seu nome: ')
```

dessa forma, o texto `'Digite seu nome: '` será exibido na tela e o programa aguardará até que o usuário entre com o texto e pressione a tecla Enter. O texto digitado pelo usuário será armazenado na variável nome.

Pode-se também trabalhar com valores inteiros e reais através da conversão do texto digitado para valores numéricos. Para isso pode-se usar as palavras-chave ***int*** e ***float*** para números inteiros e reais respectivamente. Veja um exemplo de um programa sequencial completo com a leitura de um número inteiro (ano):

```
ano_atual = 2023
nome = input('Digite seu nome: ')
ano = int(input('Digite seu ano de nascimento: '))
idade = ano_atual - ano
print('\nBom dia, %s! Você terá %d anos em 31/12/%d.' %(nome, idade, ano_atual) )
```

Uma possível saída para este programa seria a mostrada a seguir (considerando que as entradas de dados do usuário (pelo teclado) foram as que estão **destacadas**):

```
Digite seu nome: Maria
Digite seu ano de nascimento: 2001

Bom dia, Maria! Você terá 22 anos em 31/12/2023.
```