

**Instituto de
Computação**

UNIVERSIDADE ESTADUAL DE CAMPINAS



MC102 – Primeira Aula de Laboratório

Algoritmos e Programação de Computadores

Zanoni Dias

2021

Instituto de Computação

Instruções Iniciais

Avaliação

SuSy

Horários de Atendimento

Primeiro Programa

Instruções Iniciais

Instruções Iniciais

- Atividades Práticas:
 - Uma atividade prática por semana cobrindo tópicos vistos em aulas anteriores.
 - Cada atividade prática terá um prazo de entrega de pelo menos três semanas.
 - Os prazos de entrega não serão estendidos, mesmo que hajam feriados na semana de entrega.
 - Não deixe para submeter os trabalhos nos últimos dias.
- Fraudes e plágios:
 - Fraudes nas tarefas práticas poderão ser detectadas automaticamente entre todas as respostas (de todos os alunos, entre todas as turmas) ao longo do semestre.
 - A submissão de um programa que produz as saídas esperadas dos testes abertos a partir da comparação de trechos da entrada, sem de fato implementar os algoritmos solicitados nas tarefas práticas, será considerada fraude.

Avaliação

- A avaliação será realizada exclusivamente através de tarefas práticas que deverão ser implementadas pelos alunos.
- Os programas desenvolvidos serão testados com um conjunto pré-determinado de testes, subdividido em testes abertos (que podem ser acessados pelos alunos) e testes fechados (que não podem).
- A nota de cada atividade prática será proporcional ao número de testes, abertos ou fechados, que executarem corretamente.
- Juntamente com o enunciado de cada tarefa prática será indicado o peso (1, 2, 3 ou 4) da mesma.
- A média das tarefas práticas (P) será a média ponderada das notas das tarefas.

- Caso $P \geq 5$:
 - Aluno aprovado por nota e frequência com média final $F = P$.
- Caso $2,5 \leq P < 5$:
 - O aluno poderá realizar o exame composto por um subconjunto das tarefas práticas disponibilizadas ao longo do semestre.
 - O aluno poderá refazer as tarefas indicadas no período do exame ou aproveitar as notas já obtidas anteriormente naquelas tarefas (sem necessidade de refazer a tarefa).
 - A nota do exame (E) será calculada como a média ponderada das tarefas selecionadas para compor o exame. Os pesos das tarefas para fins do exame poderão ser diferentes daqueles previamente utilizados para o cálculo da média das tarefas práticas (P).
 - O cálculo da média final (F) será feita da seguinte forma:
$$F = \min\{5, (P + E)/2\}.$$
 - Caso $F \geq 5,0$ o aluno estará aprovado por nota e frequência. Caso contrário, estará reprovado por nota.

- Caso $P < 2,5$:
 - Aluno reprovado por nota com média final $F = P$.
- Observações:
 - De acordo com o Regimento Geral de Graduação, os alunos devem ter frequência maior ou igual a 75% para aprovação, mas como esta disciplina será realizada de forma online, não haverá controle de presença (será atestada 100% de presença para todos os alunos matriculados).
 - De acordo com a fórmula acima, caso um aluno seja aprovado após realizar o exame final, sua nota final será $F = 5$ (cinco).

SuSy

- Sistema de submissão e testes automáticos de programas.
- Desenvolvido pelo Prof. Tomasz Kowaltowski (Instituto de Computação - Unicamp).
- O SuSy faz detecção de plágio.
- Fraudes/plágios implicam em nota 0 (zero) no semestre.
- Nesta primeira aula prática veremos um exemplo de submissão de programa.

Horários de Atendimento

Horários de Atendimento

- Além dos horários das aulas, os alunos poderão contar com horários de atendimento de dúvidas com os monitores (PEDs e PADs).
- Cada turma terá um horário dedicado por semana.
- Além dos horários de atendimentos específicos de cada turma, haverá 35h de atendimento de dúvidas por semana com os monitores.
- Verifique os horários de atendimento em:
<https://ic.unicamp.br/~mc102/horarios>

Primeiro Programa

Primeiro Programa – Acessando o Enunciado da Atividade

1. Acesse a página do SuSy:
`https://susy.ic.unicamp.br:9999/mc102multi`
2. Clique na atividade prática “01 Aritmética com Inteiros”.
3. Na página da atividade prática clique em “Enunciado”.
4. Leia com cuidado todo o enunciado da atividade prática.
5. Na página da atividade prática clique em “Arquivos auxiliares” e realize o download do código base da atividade (lab01.py).
6. Importante: tenha certeza que o arquivo foi salvo como lab01.py (e não com outra extensão, por exemplo, lab01.py.txt). Se o arquivo foi salvo com outro nome ou extensão, renomeie para lab01.py.

Primeiro Programa – Primeira Submissão

1. Na página da atividade prática, para os campos de “Usuário” e “Senha”, informe seu RA (apenas os números) e sua senha da DAC, respectivamente.
2. Na seção “Carga de arquivos:” clique em “Choose File” e selecione o arquivo do código base que você acabou de realizar o download (lab01.py).
3. Em seguida, clique no botão “Submeter”.

Arquivos carregados:

lab01.py (lab01.py): 491 bytes

Total: 491 bytes

Fase de execução:

Teste 01: resultado incorreto

```
b = 12
a - b - c = -11
b % c = 0
```

```
| b = 12
| a - b - c = -13
<
> b % c = 0
```

Teste 02: resultado incorreto

```
b = 8
a - b - c = -7
b % c = 2
```

```
| b = 8
| a - b - c = -9
<
> b % c = 2
```

Teste 03: resultado incorreto

```
b = 3
a - b - c = -6
b % c = 3
```

```
| b = 3
| a - b - c = 0
<
> b % c = 3
```

1. Acesse a página da atividade desejada no SuSy.
2. Informe seu usuário e sua senha.
3. Clique em “Consultar”.
4. Será mostrado o relatório da sua última submissão.

1. Acesse a página da atividade desejada no SuSy.
2. Informe seu usuário e sua senha.
3. Clique em “Recuperar”.
4. Será mostrado um link para o seu último arquivo submetido.

Primeiro Programa – Primeira Correção do Código Base

1. Acesse a página do Google Cloud Shell
`https://shell.cloud.google.com`
2. Selecione a opção *Open Workspace* que aparece à direita.
3. Uma janela mostrando uma pasta com o seu nome de usuário será exibida. Basta clicar em *Open* para abri-la como um *workspace* (área de trabalho).
4. Faça upload do arquivo base (lab01.py) no Google Cloud Shell.
5. No código, preencha o seu nome e RA nas linhas indicadas.
6. Faça a primeira correção solicitada no enunciado (linha 13 do código).

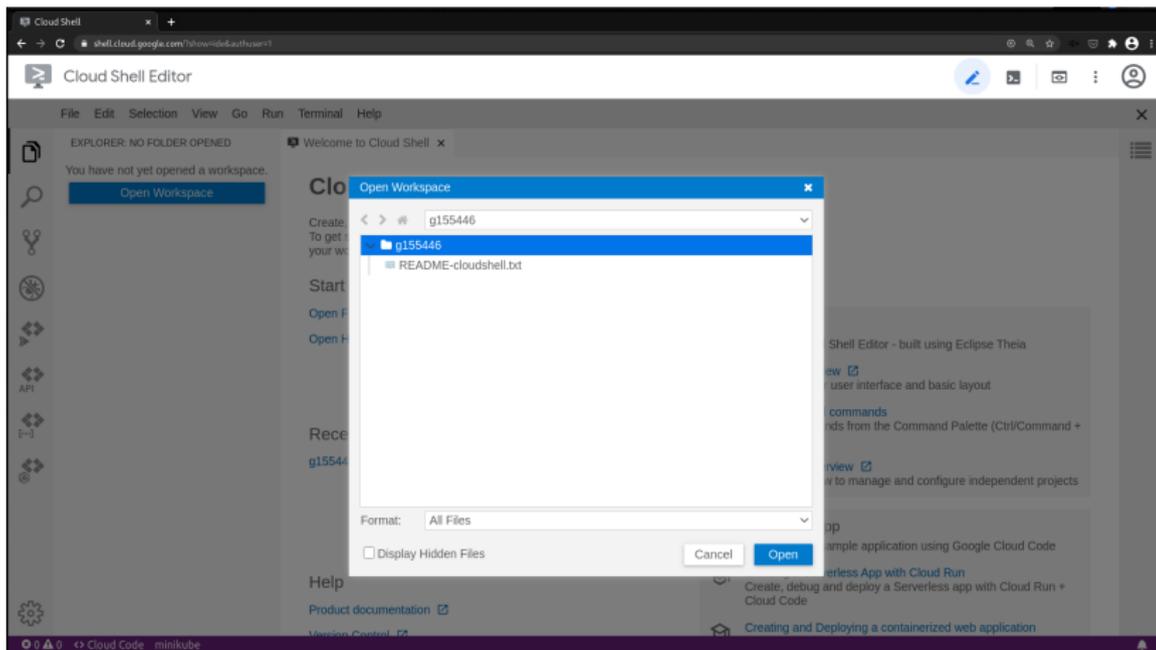
Primeiro Programa – Primeira Correção do Código Base

The screenshot displays the Cloud Shell Editor interface in a browser window. The browser address bar shows the URL `shell.cloud.google.com/showcase/auth/next`. The interface has a menu bar with `File`, `Edit`, `Selection`, `View`, `Go`, `Run`, `Terminal`, and `Help`. On the left, there is an Explorer sidebar with icons for Explorer, Search, Git, Cloud Shell, API, and Settings. The Explorer sidebar shows the message: "EXPLORER: NO FOLDER OPENED. You have not yet opened a workspace. Open Workspace". The main content area is titled "Welcome to Cloud Shell" and contains the following sections:

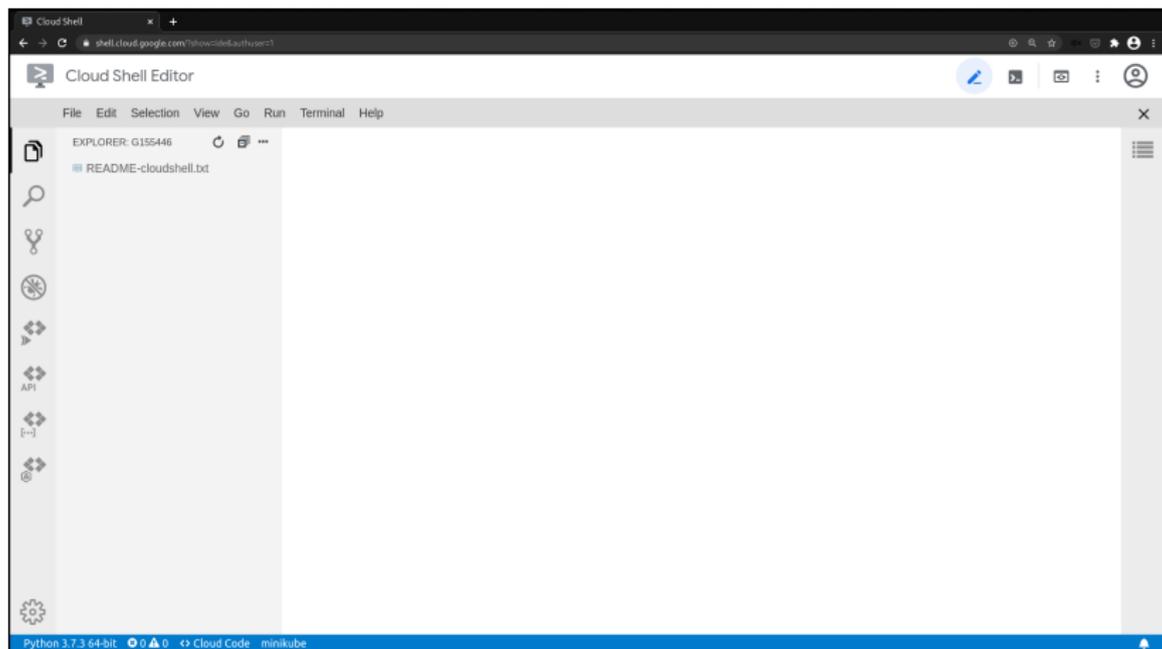
- Cloud Shell Editor**: A heading followed by the text: "Create, build and deploy your cloud-native applications in an online editor. To get started, open your [home folder](#) or one of the options below to load your workspace."
- Start**: A section with two links: [Open Folder...](#) and [Open Home Workspace](#).
- Recent**: A section with one entry: `g155446 /home/g155446`.
- Help**: A section with two links: [Product documentation](#) and [Version Control](#).
- Learn**: A section with a sub-section **Editor Overview** containing three links: [Interface overview](#), [Find and run all commands](#), and [Workspace overview](#).
- Create a sample app**: A section with the text "Create and Deploy a sample application using Google Cloud Code" and three links: [Creating a Serverless App with Cloud Run](#), [Creating and Deploying a containerized web application](#), and [Creating and Deploying a Serverless App with Cloud Run + Cloud Code](#).

The status bar at the bottom shows "Cloud Code" and the user name "minikube".

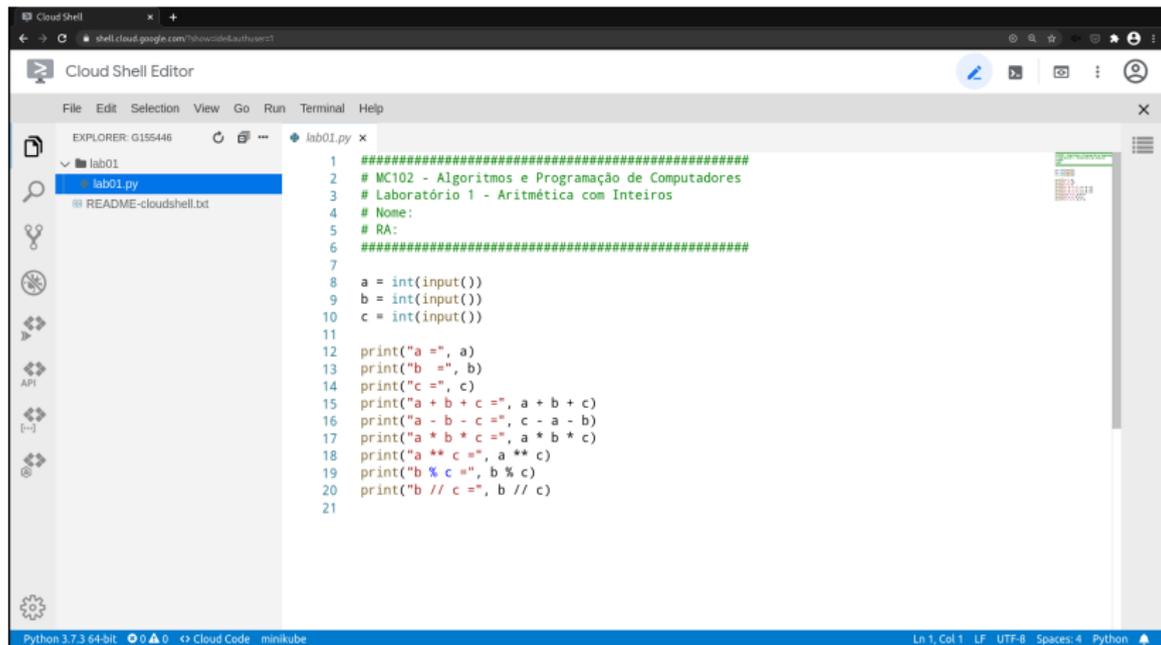
Primeiro Programa – Primeira Correção do Código Base



Primeiro Programa – Primeira Correção do Código Base



Primeiro Programa – Primeira Correção do Código Base

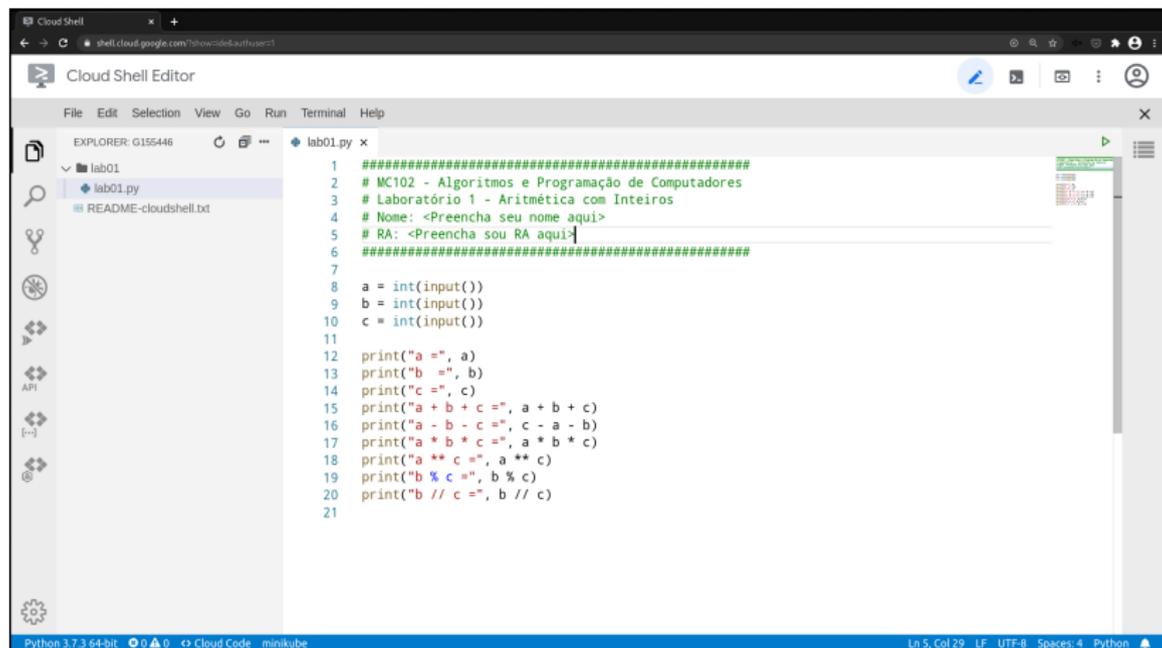


The screenshot shows a Cloud Shell Editor window with a menu bar (File, Edit, Selection, View, Go, Run, Terminal, Help) and a toolbar. The Explorer on the left shows a file named lab01.py. The main editor area contains the following code:

```
1 #####
2 # MC102 - Algoritmos e Programação de Computadores
3 # Laboratório 1 - Aritmética com Inteiros
4 # Nome:
5 # RA:
6 #####
7
8 a = int(input())
9 b = int(input())
10 c = int(input())
11
12 print("a =", a)
13 print("b =", b)
14 print("c =", c)
15 print("a + b + c =", a + b + c)
16 print("a - b - c =", c - a - b)
17 print("a * b * c =", a * b * c)
18 print("a ** c =", a ** c)
19 print("b % c =", b % c)
20 print("b // c =", b // c)
21
```

The status bar at the bottom indicates Python 3.7.3 64-bit, Cloud Code, minikube, Ln 1, Col 1, LF, UTF-8, Spaces: 4, Python.

Primeiro Programa – Primeira Correção do Código Base

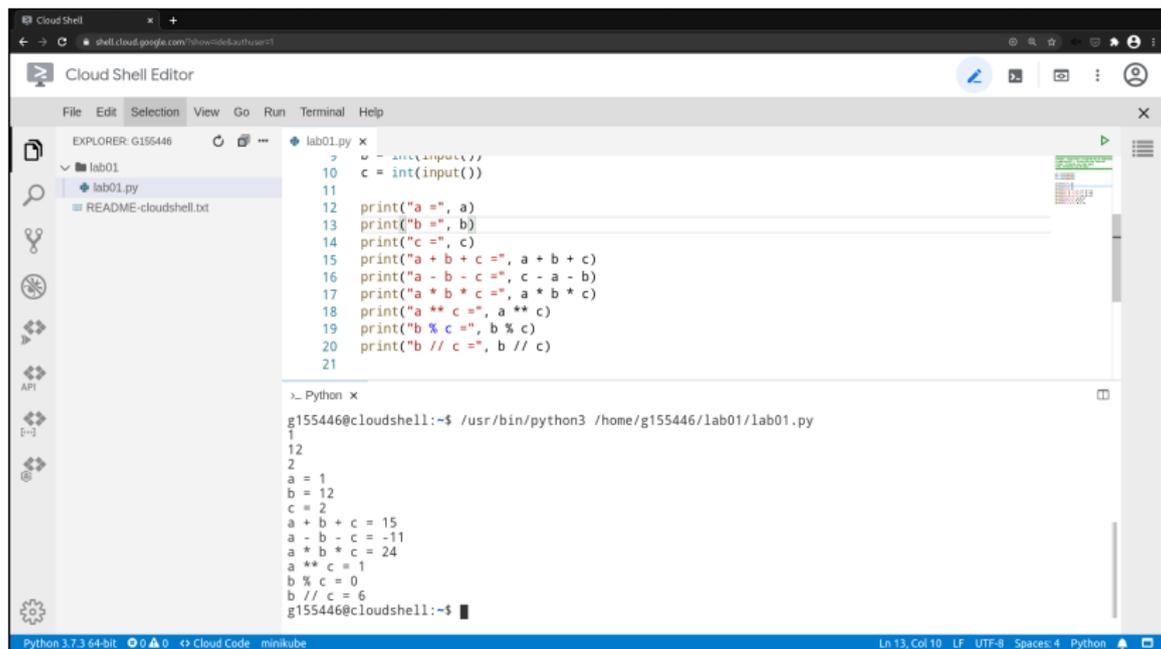


```
Cloud Shell Editor
File Edit Selection View Go Run Terminal Help
EXPLORER: G155446
lab01
  lab01.py
  README-cloudshell.txt
lab01.py x
1 #####
2 # MC102 - Algoritmos e Programação de Computadores
3 # Laboratório 1 - Aritmética com Inteiros
4 # Nome: <Preencha seu nome aqui>
5 # RA: <Preencha sou RA aqui>
6 #####
7
8 a = int(input())
9 b = int(input())
10 c = int(input())
11
12 print("a =", a)
13 print("b =", b)
14 print("c =", c)
15 print("a + b + c =", a + b + c)
16 print("a - b - c =", c - a - b)
17 print("a * b * c =", a * b * c)
18 print("a ** c =", a ** c)
19 print("b % c =", b % c)
20 print("b // c =", b // c)
21
Python 3.7.3 64-bit Cloud Code minikube Ln 5, Col 29 LF UTF-8 Spaces: 4 Python
```

Primeiro Programa – Primeira Correção do Código Base

1. Clique na seta verde no canto superior esquerdo para rodar o programa.
2. Em seguida, digite no terminal (campo abaixo do código) as entradas para o seu programa conforme mostrado no enunciado da atividade.
3. Baixe o arquivo lab01.py e submeta no susy.
4. Sua submissão ainda deve gerar um relatório com “resultado incorreto” para todos os casos de teste.

Primeiro Programa – Primeira Correção do Código Base



The screenshot displays the Cloud Shell Editor interface. The Explorer on the left shows a file named 'lab01.py' under the directory 'lab01'. The main editor area contains the following Python code:

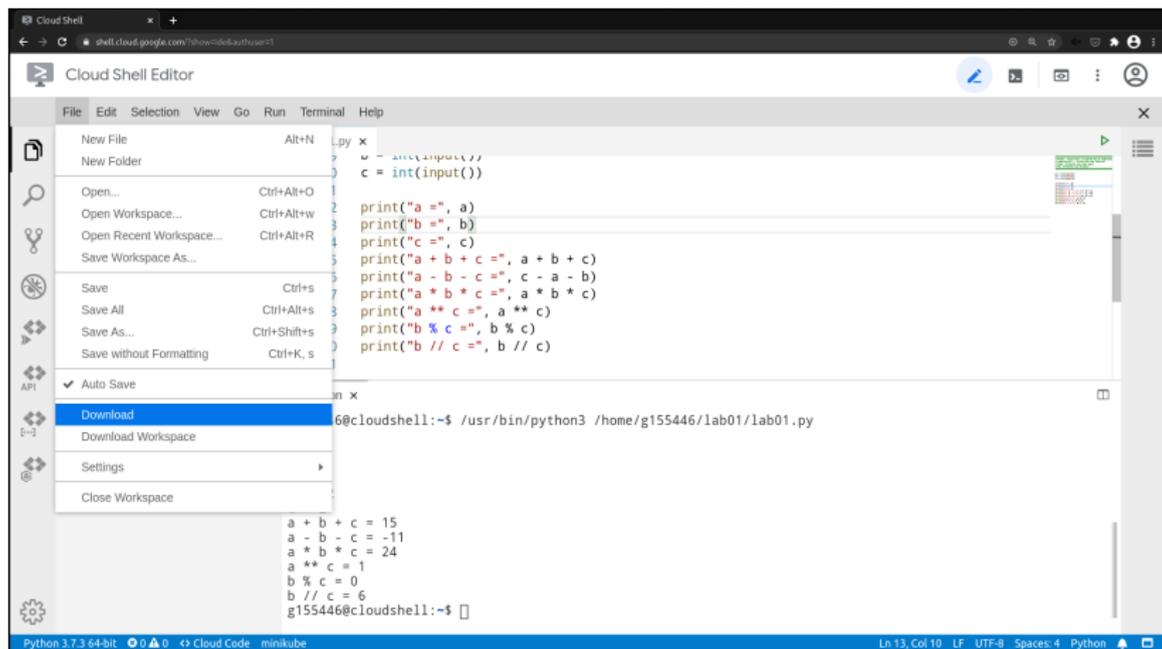
```
1 a = int(input())
2 b = int(input())
3 c = int(input())
4
5
6
7
8
9
10 c = int(input())
11
12 print("a =", a)
13 print("b =", b)
14 print("c =", c)
15 print("a + b + c =", a + b + c)
16 print("a - b - c =", c - a - b)
17 print("a * b * c =", a * b * c)
18 print("a ** c =", a ** c)
19 print("b % c =", b % c)
20 print("b // c =", b // c)
21
```

Below the code editor, the terminal shows the execution of the script:

```
Python x
g155446@cloudshell:~$ /usr/bin/python3 /home/g155446/lab01/lab01.py
1
2
a = 1
b = 12
c = 2
a + b + c = 15
a - b - c = -11
a * b * c = 24
a ** c = 1
b % c = 0
b // c = 6
g155446@cloudshell:~$
```

The status bar at the bottom indicates 'Python 3.7.3 64-bit', 'Cloud Code', 'minikube', 'Ln 13, Col 10', 'LF', 'UTF-8', 'Spaces: 4', and 'Python'.

Primeiro Programa – Primeira Correção do Código Base



The screenshot shows the Cloud Shell Editor interface. A menu is open over the 'Download' option. The Python code in the background is as follows:

```
1 # -*- coding: utf-8 -*-
2
3 c = int(input())
4
5 print("a =", a)
6 print("b =", b)
7 print("c =", c)
8 print("a + b + c =", a + b + c)
9 print("a - b - c =", c - a - b)
10 print("a * b * c =", a * b * c)
11 print("a ** c =", a ** c)
12 print("b % c =", b % c)
13 print("b // c =", b // c)
```

The terminal output shows the results of the script execution:

```
6@cloudshell:~$ /usr/bin/python3 /home/g155446/lab01/lab01.py
a + b + c = 15
a - b - c = -11
a * b * c = 24
a ** c = 1
b % c = 0
b // c = 6
g155446@cloudshell:~$
```

The status bar at the bottom indicates: Python 3.7.3 64-bit, Cloud Code, minikube, Ln 13, Col 10, LF, UTF-8, Spaces: 4, Python.

Arquivos carregados:

lab01.py (lab01.py): 490 bytes

Total: 490 bytes

Fase de execução:

Teste 01: resultado incorreto

```
a - b - c = -11
b % c = 0
```

```
| a - b - c = -13
<
> b % c = 0
```

Teste 02: resultado incorreto

```
a - b - c = -7
b % c = 2
```

```
| a - b - c = -9
<
> b % c = 2
```

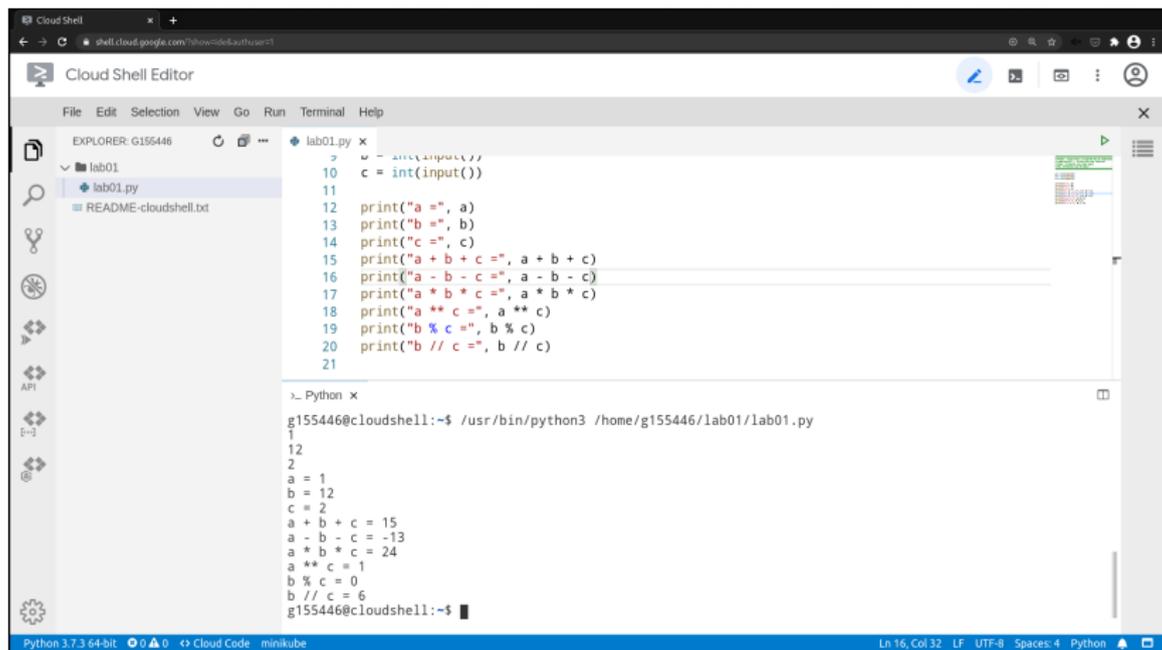
Teste 03: resultado incorreto

```
a - b - c = -6
b % c = 3
```

```
| a - b - c = 0
<
> b % c = 3
```

1. No Google Cloud Shell, faça a segunda correção solicitada no enunciado (linha 16 do código).
2. Em seguida, teste novamente seu programa com as entradas fornecidas na atividade.
3. Baixe o arquivo lab01.py e submeta no susy.
4. Sua submissão ainda deve gerar um relatório com “resultado incorreto” para todos os casos de teste.

Primeiro Programa – Segunda Correção do Código Base



The screenshot displays the Cloud Shell Editor interface. The top bar shows the browser address bar with the URL `shell.cloud.google.com/show?id=cautbzsm7`. The editor title is "Cloud Shell Editor". The menu bar includes "File", "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help".

The Explorer pane on the left shows the file structure for workspace G155446, containing a folder "lab01" with files "lab01.py" and "README-cloudshell.txt".

The main editor area shows the code in `lab01.py`:

```
9 # -*- coding: utf-8 -*-
10 c = int(input())
11
12 print("a =", a)
13 print("b =", b)
14 print("c =", c)
15 print("a + b + c =", a + b + c)
16 print("a - b - c =", a - b - c)
17 print("a * b * c =", a * b * c)
18 print("a ** c =", a ** c)
19 print("b % c =", b % c)
20 print("b // c =", b // c)
21
```

The Terminal pane shows the execution of the script:

```
Python 3
g155446@cloudshell:~$ /usr/bin/python3 /home/g155446/lab01/lab01.py
1
2
a = 1
b = 12
c = 2
a + b + c = 15
a - b - c = -13
a * b * c = 24
a ** c = 1
b % c = 0
b // c = 6
g155446@cloudshell:~$
```

The status bar at the bottom indicates "Python 3.7.3 64-bit", "Cloud Code", "minikube", "Ln 16, Col 32", "LF", "UTF-8", "Spaces: 4", and "Python".

Arquivos carregados:

lab01.py (lab01.py): 490 bytes

Total: 490 bytes

Fase de execução:

Teste 01: resultado incorreto

b % c = 0

<
> b % c = 0

Teste 02: resultado incorreto

b % c = 2

<
> b % c = 2

Teste 03: resultado incorreto

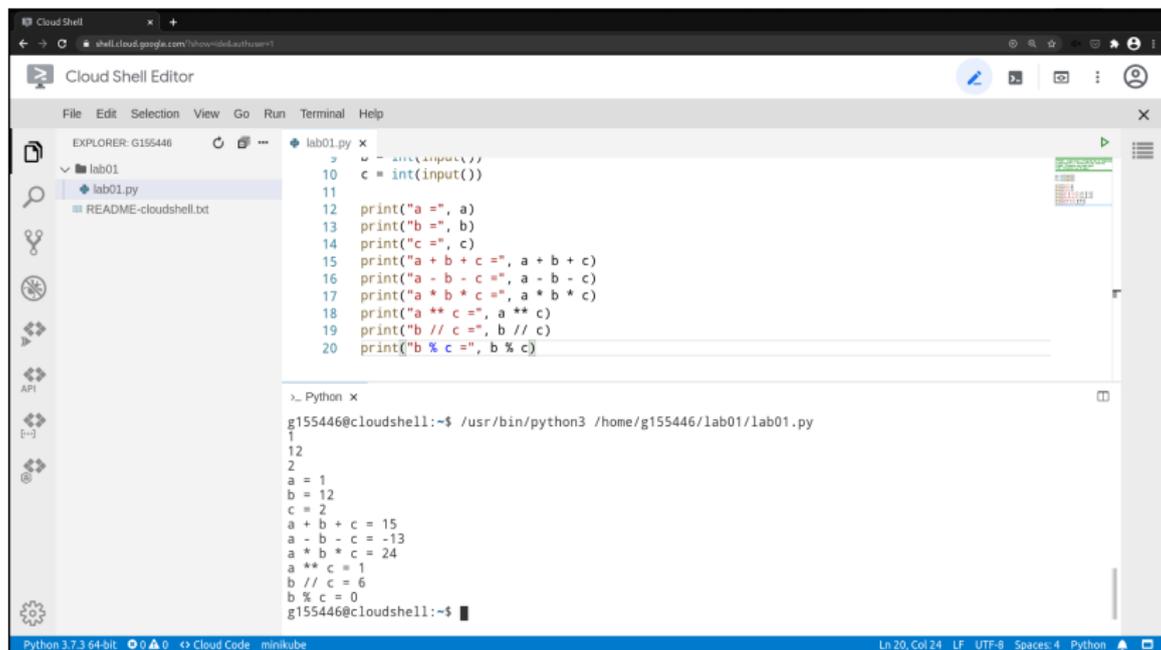
b % c = 3

<
> b % c = 3

Primeiro Programa – Terceira Correção do Código Base

1. Voltando ao Google Cloud Shell, faça a terceira correção solicitada no enunciado (linhas 19 e 20 do código).
2. Teste novamente seu programa com as entradas fornecidas na atividade.
3. Baixe o arquivo lab01.py e submeta no susy.
4. Nessa submissão o relatório gerado deve indicar “resultado correto” para todos os casos de teste.

Primeiro Programa – Terceira Correção do Código Base



The screenshot displays the Cloud Shell Editor interface. The Explorer on the left shows a file named 'lab01.py' under the directory 'lab01'. The main editor area contains the following Python code:

```
10 c = int(input())
11
12 print("a =", a)
13 print("b =", b)
14 print("c =", c)
15 print("a + b + c =", a + b + c)
16 print("a - b - c =", a - b - c)
17 print("a * b * c =", a * b * c)
18 print("a ** c =", a ** c)
19 print("b // c =", b // c)
20 print("b % c =", b % c)
```

Below the code editor, the terminal shows the execution of the script:

```
>_ Python x
g155446@cloudshell:~$ /usr/bin/python3 /home/g155446/lab01/lab01.py
1
12
2
a = 1
b = 12
c = 2
a + b + c = 15
a - b - c = -13
a * b * c = 24
a ** c = 1
b // c = 6
b % c = 0
g155446@cloudshell:~$
```

The status bar at the bottom indicates the environment is Python 3.7.3 64-bit, running on Cloud Code minikube, with the cursor at line 20, column 24.

Arquivos carregados:

lab01.py (lab01.py): 490 bytes

Total: 490 bytes

Fase de execução:

```
Teste 01: resultado correto
Teste 02: resultado correto
Teste 03: resultado correto
Teste 04: resultado correto
Teste 05: resultado correto
Teste 06: resultado correto
Teste 07: resultado correto
Teste 08: resultado correto
Teste 09: resultado correto
Teste 10: resultado correto
Teste 11: resultado correto
Teste 12: resultado correto
Teste 13: resultado correto
Teste 14: resultado correto
Teste 15: resultado correto
Teste 16: resultado correto
Teste 17: resultado correto
Teste 18: resultado correto
Teste 19: resultado correto
Teste 20: resultado correto
```

- São permitidas no máximo 20 submissões no SuSy para cada atividade prática.
- Utilize o sistema SuSy com o seu RA (apenas números) e com a senha que você utiliza para fazer acesso ao sistema da DAC.
- Para avaliação, será considerado apenas o resultado da última submissão.
- Você deve seguir com cuidado as instruções de submissão descritas no enunciado.
- Não use o SuSy para testar o seu programa: sempre teste seu programa com os casos de testes abertos, antes de submeter o seu programa para avaliação no SuSy.
- Para mais informações, visite o site da disciplina:
<https://ic.unicamp.br/~mc102>