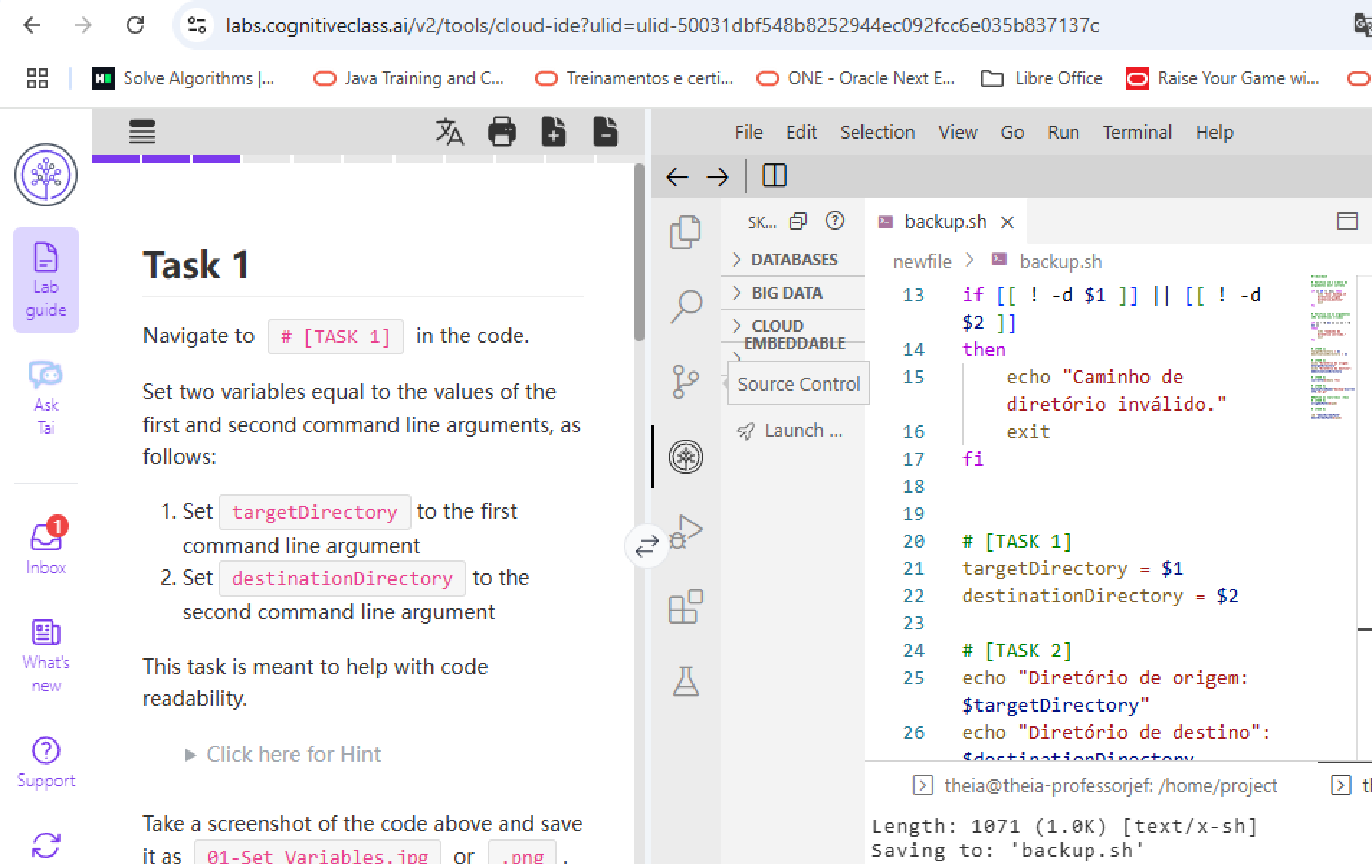


Task 1

Navigate to `# [TASK 1]` in the code.

Set two variables equal to the values of the first and second command line arguments, as follows:

1. Set `targetDirectory` to the first command line argument
2. Set `destinationDirectory` to the second command line argument

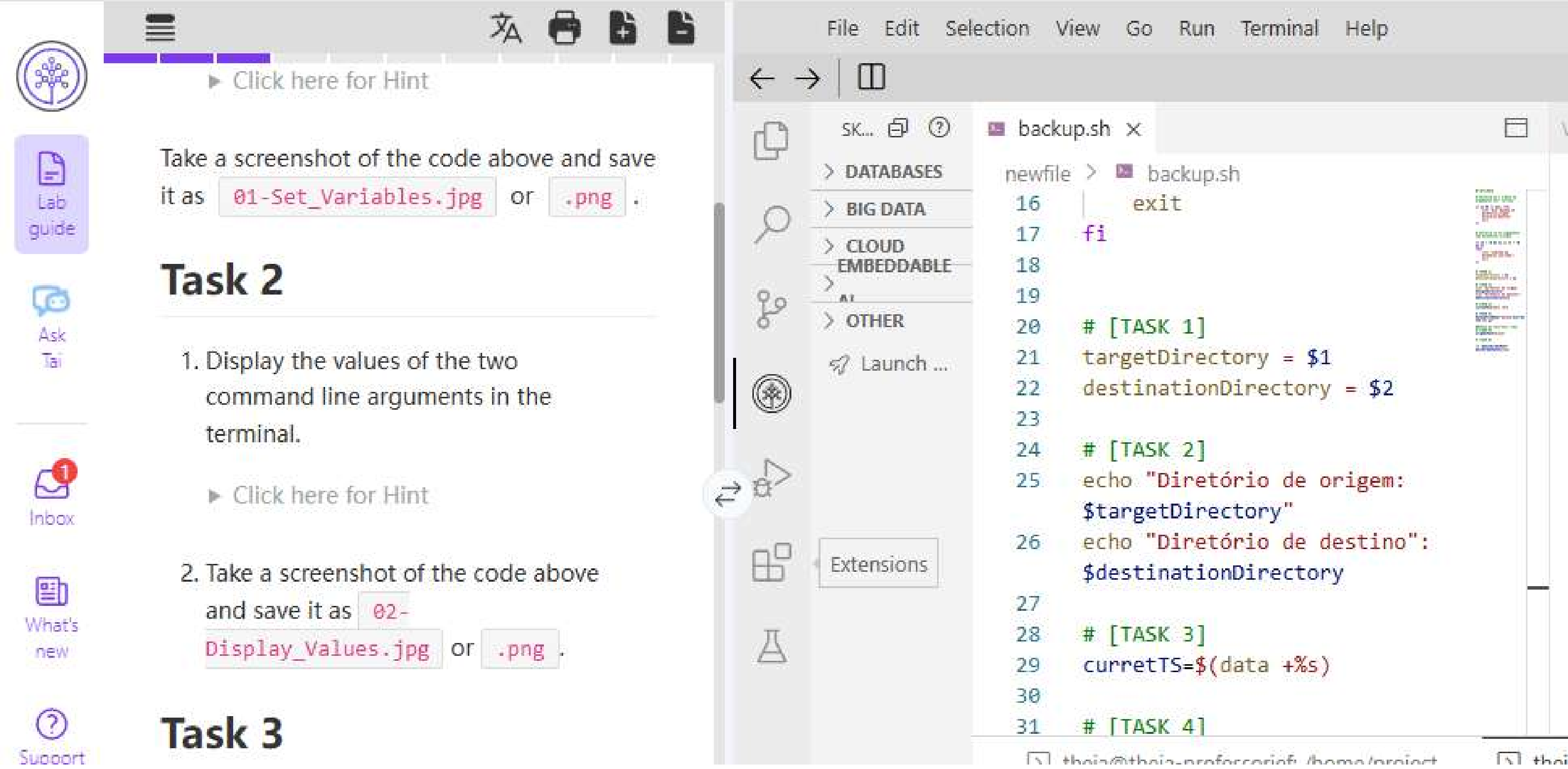
This task is meant to help with code readability.

▶ Click here for Hint

Take a screenshot of the code above and save it as `01-Set Variables.jpg` or `.png` .

```
if [[ ! -d $1 ]] || [[ ! -d $2 ]]
then
    echo "Caminho de diretório inválido."
    exit
fi


# [TASK 1]
targetDirectory = $1
destinationDirectory = $2

# [TASK 2]
echo "Diretório de origem: $targetDirectory"
echo "Diretório de destino": $destinationDirectory
```

```
Length: 1071 (1.0K) [text/x-sh]
Saving to: 'backup.sh'
```



▸ Click here for Hint

Take a screenshot of the code above and save it as `01-Set_Variables.jpg` or `.png` .

Task 2

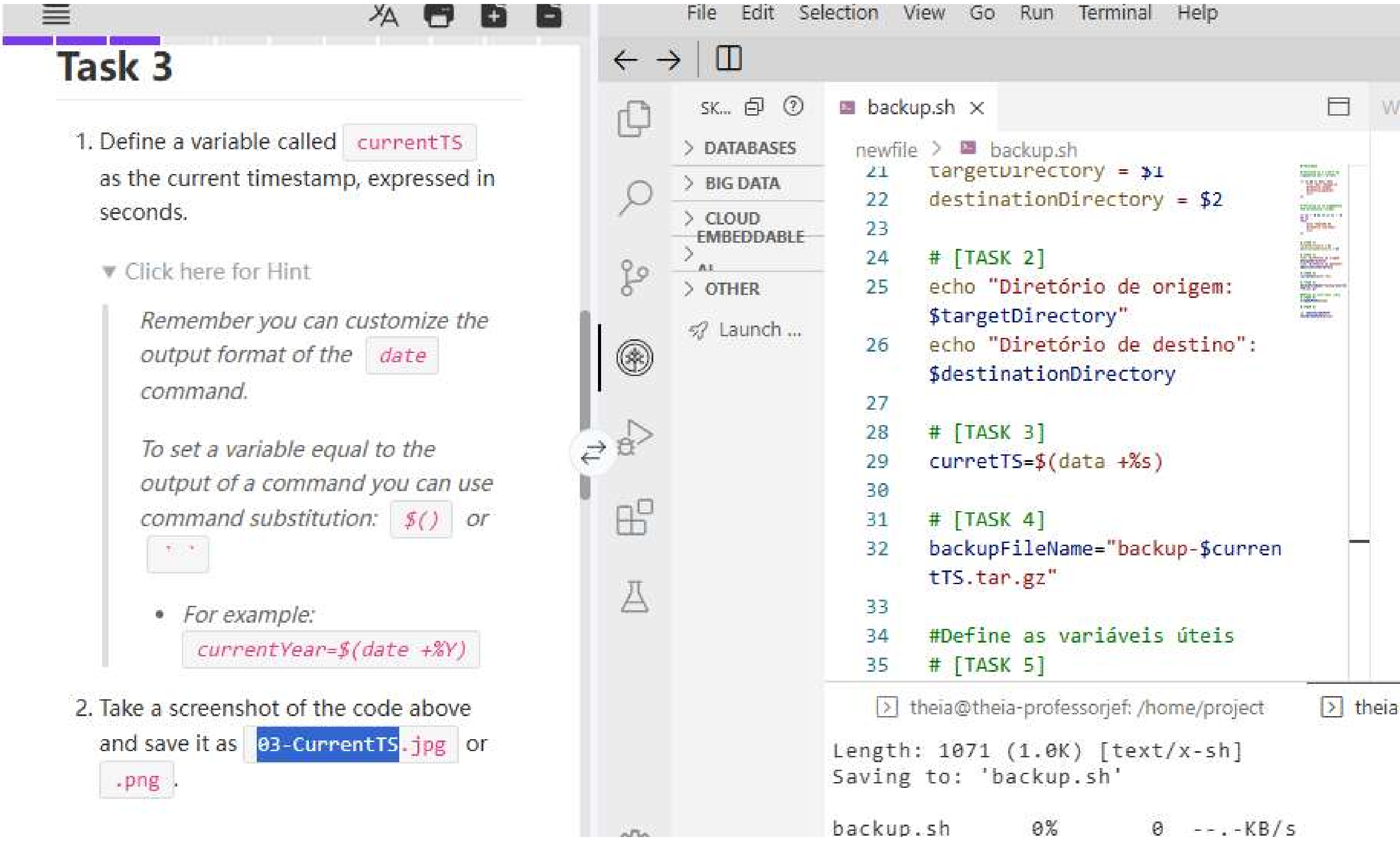
1. Display the values of the two command line arguments in the terminal.

   ▸ Click here for Hint

2. Take a screenshot of the code above and save it as `02-Display_Values.jpg` or `.png`.

Task 3

```
    exit
fi


# [TASK 1]
targetDirectory = $1
destinationDirectory = $2

# [TASK 2]
echo "Diretório de origem: $targetDirectory"
echo "Diretório de destino": $destinationDirectory

# [TASK 3]
curretTS=$(data +%s)

# [TASK 4]
```



Task 3

1. Define a variable called `currentTS` as the current timestamp, expressed in seconds.

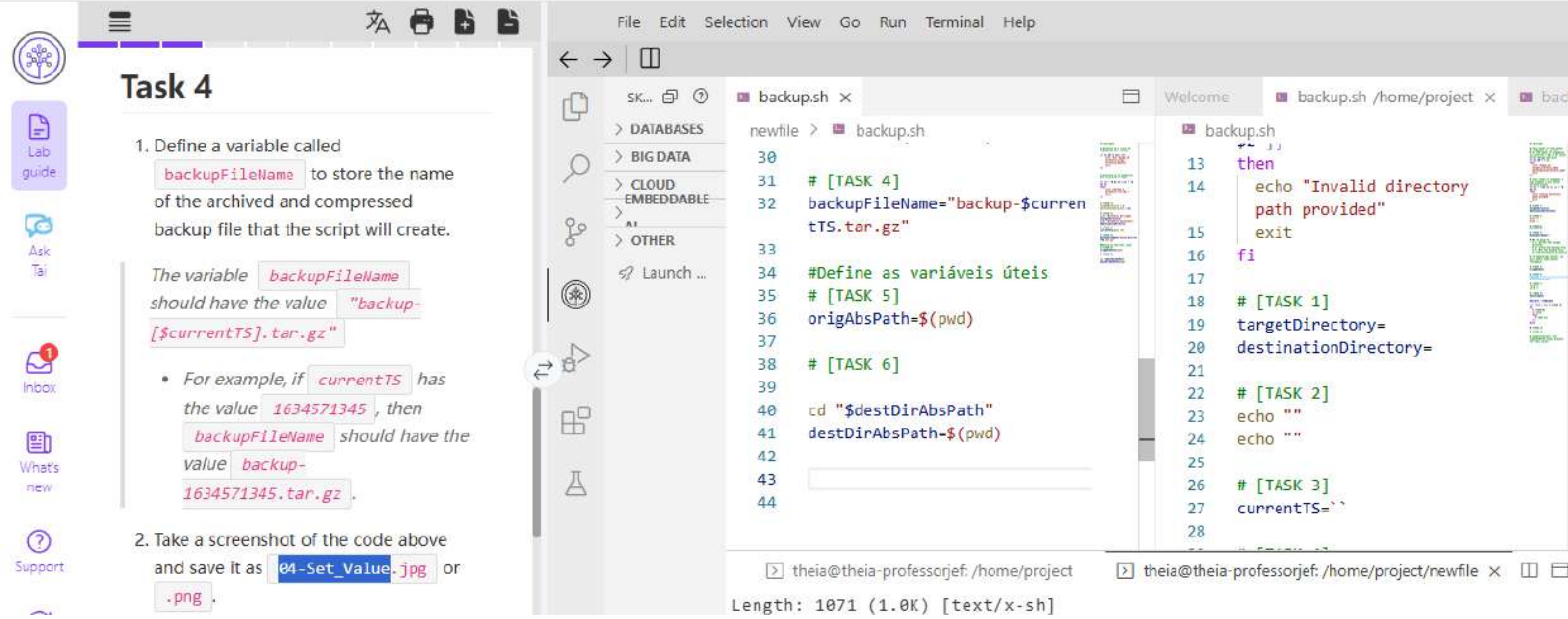
   ▼ Click here for Hint

   > *Remember you can customize the output format of the `date` command.*
   >
   > *To set a variable equal to the output of a command you can use command substitution: `$()` or `` ` ` ``*
   >
   > - *For example: `currentYear=$(date +%Y)`*

2. Take a screenshot of the code above and save it as `03-CurrentTS.jpg` or `.png`.

```
targetDirectory = $1
destinationDirectory = $2

# [TASK 2]
echo "Diretório de origem:
$targetDirectory"
echo "Diretório de destino":
$destinationDirectory

# [TASK 3]
curretTS=$(data +%s)

# [TASK 4]
backupFileName="backup-$curren
tTS.tar.gz"

#Define as variáveis úteis
# [TASK 5]
```

```
Length: 1071 (1.0K) [text/x-sh]
Saving to: 'backup.sh'

backup.sh       0%        0  --.-KB/s
```



Task 4

1. Define a variable called `backupFileName` to store the name of the archived and compressed backup file that the script will create.

> *The variable `backupFileName` should have the value `"backup-[$currentTS].tar.gz"`*
>
> - *For example, if `currentTS` has the value `1634571345`, then `backupFileName` should have the value `backup-1634571345.tar.gz`.*

2. Take a screenshot of the code above and save it as `04-Set_Value.jpg` or `.png`.

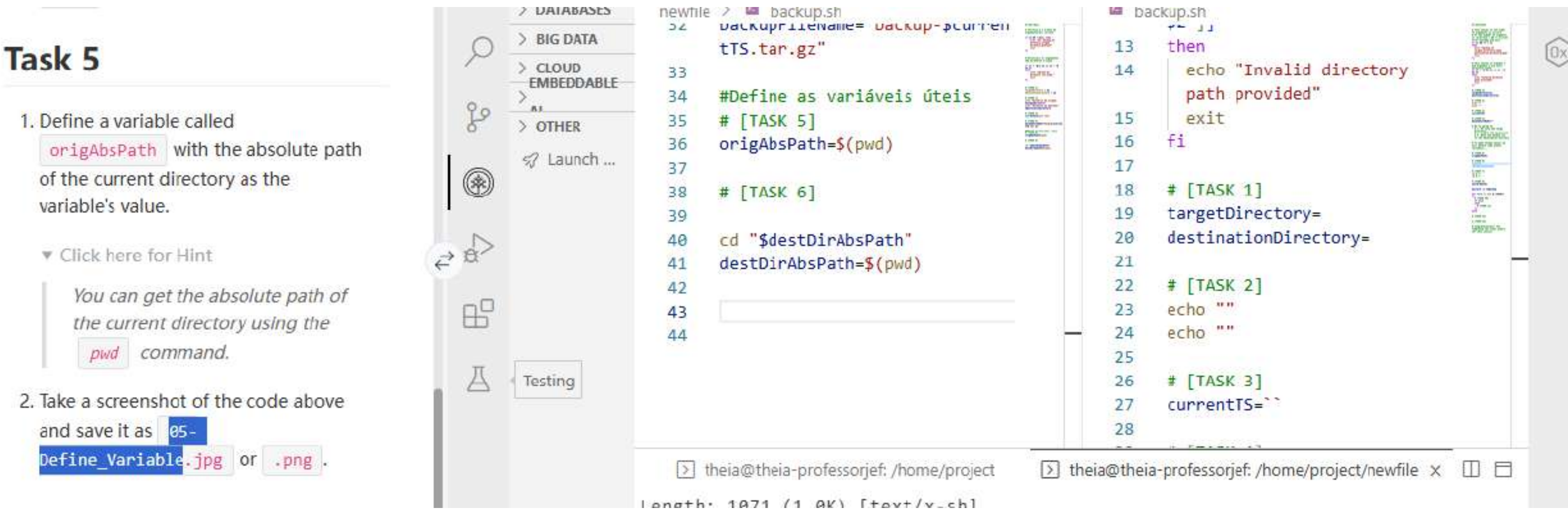
```
# [TASK 4]
backupFileName="backup-$currentTS.tar.gz"

#Define as variáveis úteis
# [TASK 5]
origAbsPath=$(pwd)

# [TASK 6]

cd "$destDirAbsPath"
destDirAbsPath-$(pwd)


```

```
then
  echo "Invalid directory path provided"
  exit
fi

# [TASK 1]
targetDirectory=
destinationDirectory=

# [TASK 2]
echo ""
echo ""

# [TASK 3]
currentTS=``

```

theia@theia-professorjef: /home/project

theia@theia-professorjef: /home/project/newfile

```
Length: 1071 (1.0K) [text/x-sh]
```



DATABASES
BIG DATA
CLOUD EMBEDDABLE
AI
OTHER
Launch ...
Testing
newfile > backup.sh
backup.sh
theia@theia-professorjef: /home/project
theia@theia-professorjef: /home/project/newfile

Task 5

1. Define a variable called `origAbsPath` with the absolute path of the current directory as the variable's value.

   ▾ Click here for Hint

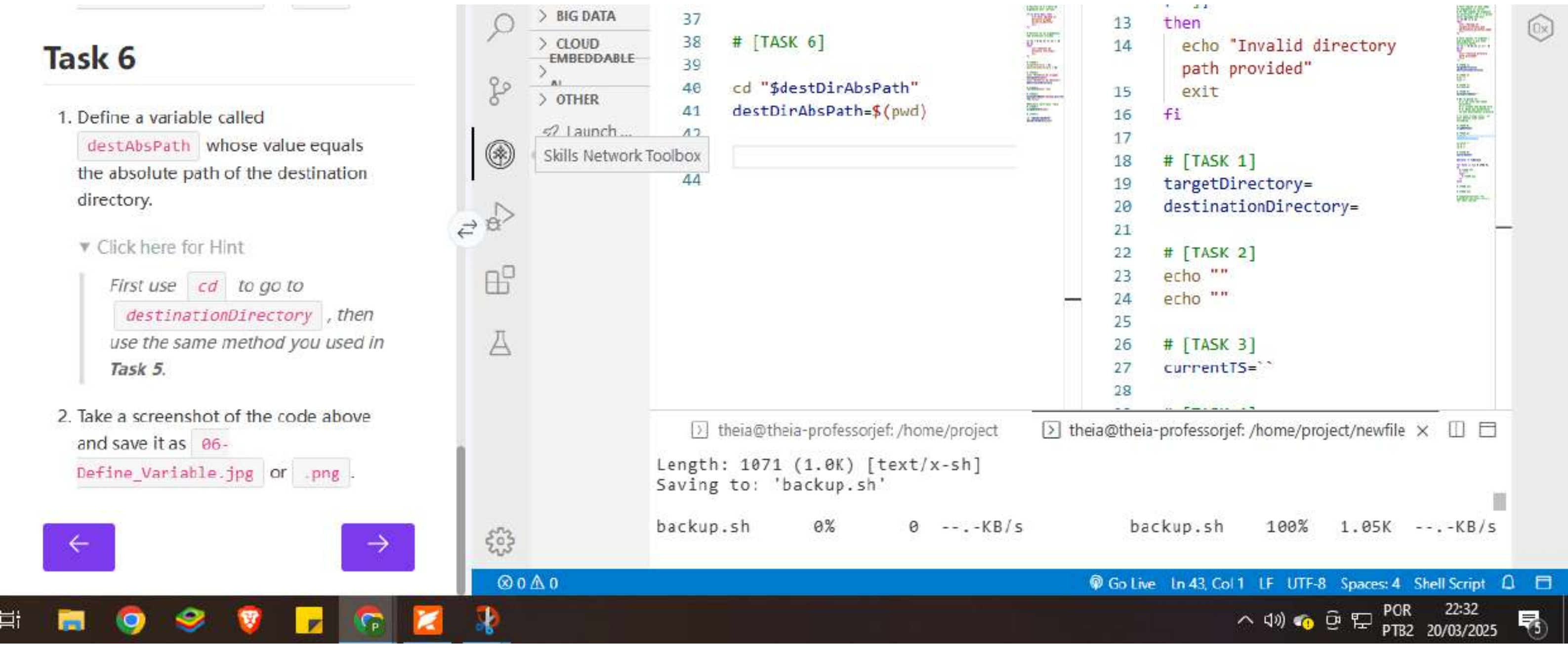
   > *You can get the absolute path of the current directory using the `pwd` command.*

2. Take a screenshot of the code above and save it as `05-Define_Variable.jpg` or `.png`.

```
backupFileName="backup-$currentTS.tar.gz"

#Define as variáveis úteis
# [TASK 5]
origAbsPath=$(pwd)

# [TASK 6]

cd "$destDirAbsPath"
destDirAbsPath=$(pwd)


```

```
then
  echo "Invalid directory path provided"
  exit
fi

# [TASK 1]
targetDirectory=
destinationDirectory=

# [TASK 2]
echo ""
echo ""

# [TASK 3]
currentTS=``

```

```
Length: 1071 (1.0K) [text/x-sh]
```



Task 6

1. Define a variable called `destAbsPath` whose value equals the absolute path of the destination directory.

   ▼ Click here for Hint

   > *First use `cd` to go to `destinationDirectory`, then use the same method you used in **Task 5**.*

2. Take a screenshot of the code above and save it as `06-Define_Variable.jpg` or `.png`.

```
# [TASK 6]

cd "$destDirAbsPath"
destDirAbsPath=$(pwd)
```

```
then
    echo "Invalid directory path provided"
    exit
fi

# [TASK 1]
targetDirectory=
destinationDirectory=

# [TASK 2]
echo ""
echo ""

# [TASK 3]
currentTS=``
```

theia@theia-professorjef: /home/project
theia@theia-professorjef: /home/project/newfile

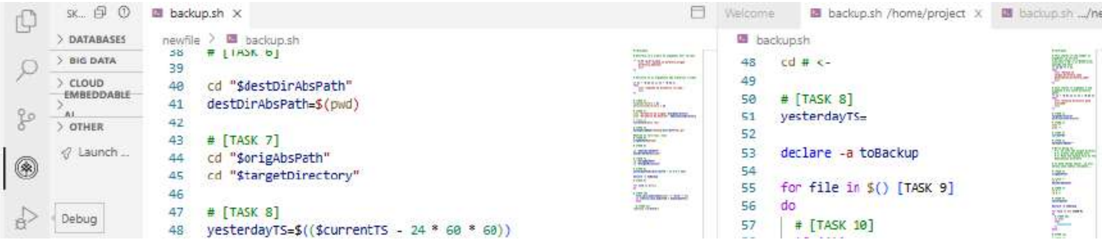
```
Length: 1071 (1.0K) [text/x-sh]
Saving to: 'backup.sh'

backup.sh        0%        0  --.-KB/s          backup.sh     100%   1.05K  --.-KB/s
```

Task 7

1. Change directories from the current working directory to the target directory `targetDirectory`.

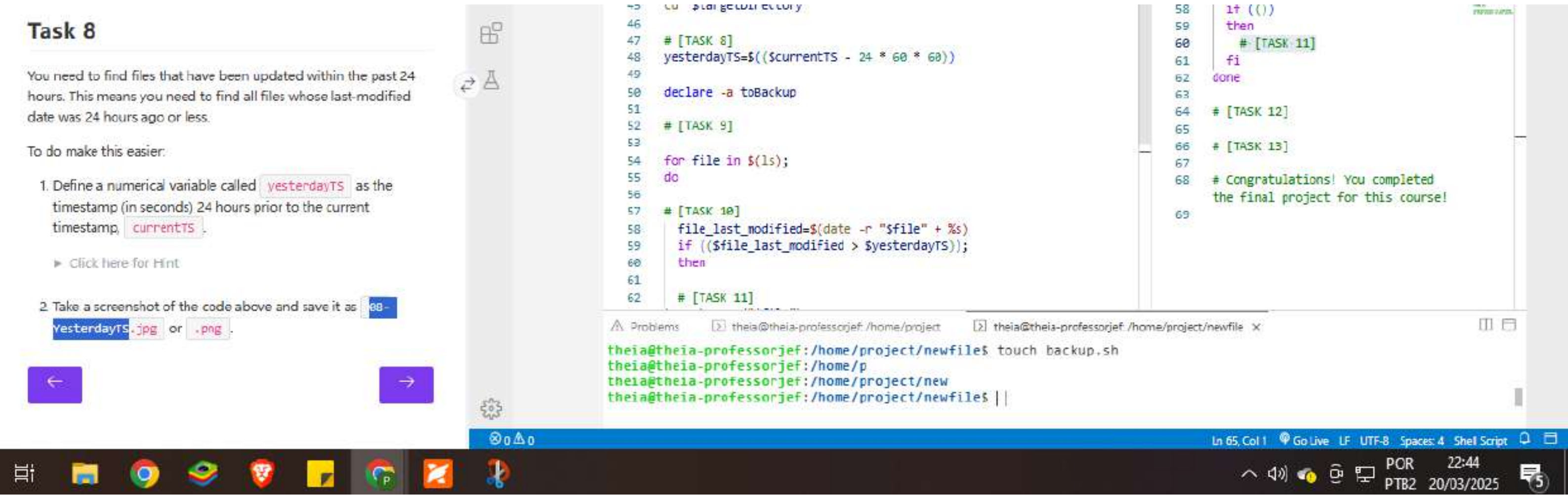
   ▸ Click here for Hint

2. Take a screenshot of the code above and save it as `07-Change_Directory.jpg` or `.png`.



```
# [TASK 6]

cd "$destDirAbsPath"
destDirAbsPath=$(pwd)

# [TASK 7]
cd "$origAbsPath"
cd "$targetDirectory"

# [TASK 8]
yesterdayTS=$(($currentTS - 24 * 60 * 60))
```

```
cd # <-

# [TASK 8]
yesterdayTS=

declare -a toBackup

for file in $() [TASK 9]
do
  # [TASK 10]
```



Task 8

You need to find files that have been updated within the past 24 hours. This means you need to find all files whose last-modified date was 24 hours ago or less.

To do make this easier:

1. Define a numerical variable called `yesterdayTS` as the timestamp (in seconds) 24 hours prior to the current timestamp, `currentTS`.

   ▸ Click here for Hint

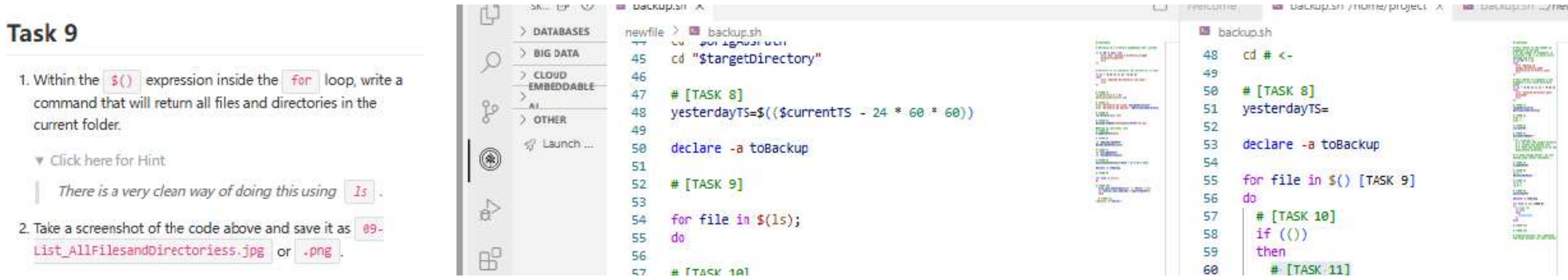
2. Take a screenshot of the code above and save it as `08-yesterdayTS.jpg` or `.png`.

```
cd $targetDirectory

# [TASK 8]
yesterdayTS=$(($currentTS - 24 * 60 * 60))

declare -a toBackup

# [TASK 9]

for file in $(ls);
do

# [TASK 10]
  file_last_modified=$(date -r "$file" + %s)
  if (($file_last_modified > $yesterdayTS));
  then

  # [TASK 11]
```

```
  if (())
  then
    # [TASK 11]
  fi
done

# [TASK 12]

# [TASK 13]

# Congratulations! You completed the final project for this course!
```

Problems
theia@theia-professorjef: /home/project
theia@theia-professorjef: /home/project/newfile

```
theia@theia-professorjef:/home/project/newfile$ touch backup.sh
theia@theia-professorjef:/home/p
theia@theia-professorjef:/home/project/new
theia@theia-professorjef:/home/project/newfile$ ||
```



Task 9

1. Within the `$()` expression inside the `for` loop, write a command that will return all files and directories in the current folder.

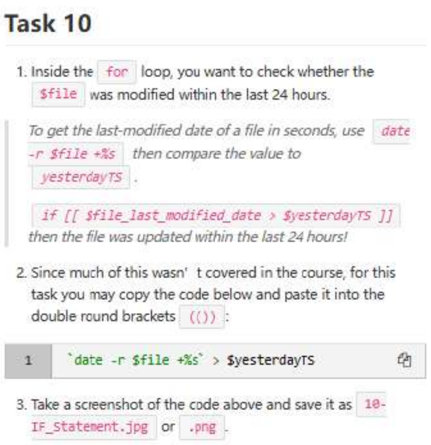
   ▼ Click here for Hint

   > *There is a very clean way of doing this using* `ls`.

2. Take a screenshot of the code above and save it as `09-List_AllFilesandDirectoriess.jpg` or `.png`.

```
cd "$origAbsPath"
cd "$targetDirectory"

# [TASK 8]
yesterdayTS=$(($currentTS - 24 * 60 * 60))

declare -a toBackup

# [TASK 9]

for file in $(ls);
do

# [TASK 10]
```

```
cd # <-

# [TASK 8]
yesterdayTS=

declare -a toBackup

for file in $() [TASK 9]
do
  # [TASK 10]
  if (())
  then
    # [TASK 11]
```



Task 10

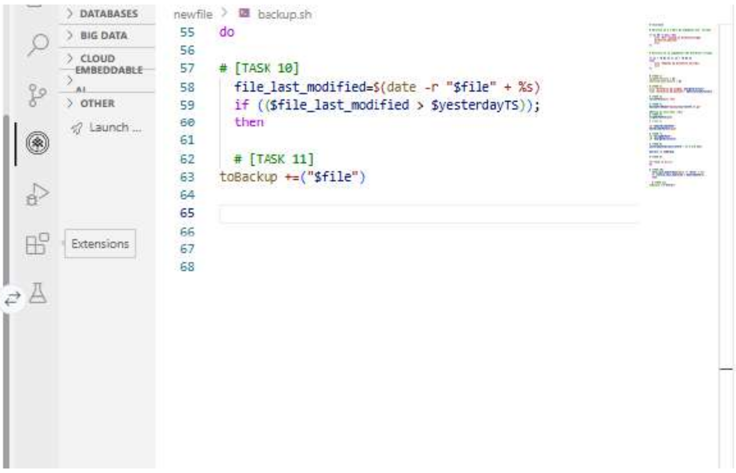
1. Inside the `for` loop, you want to check whether the `$file` was modified within the last 24 hours.

> *To get the last-modified date of a file in seconds, use* `date -r $file +%s` *then compare the value to* `yesterdayTS`.
>
> `if [[ $file_last_modified_date > $yesterdayTS ]]` *then the file was updated within the last 24 hours!*

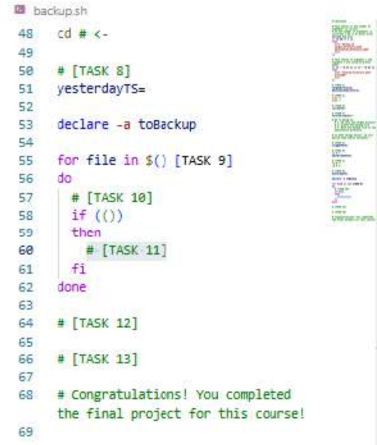
2. Since much of this wasn' t covered in the course, for this task you may copy the code below and paste it into the double round brackets `(())`:

```
`date -r $file +%s` > $yesterdayTS
```

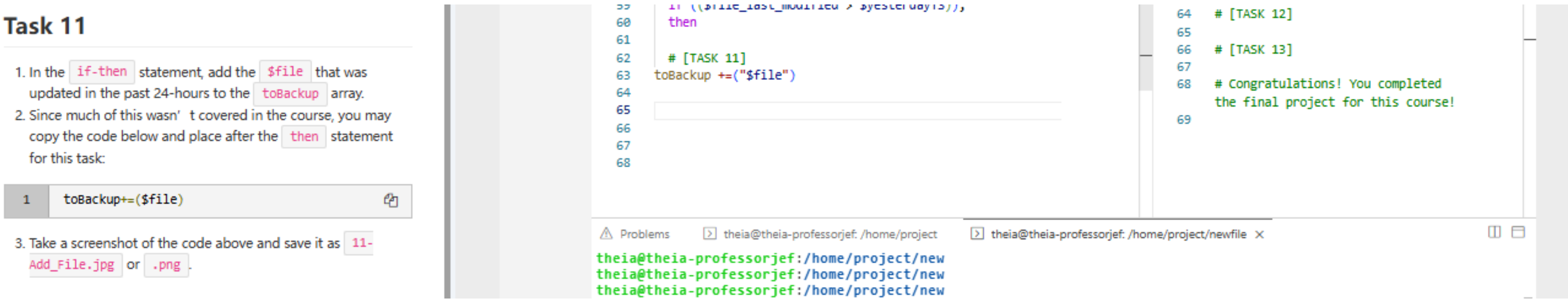
3. Take a screenshot of the code above and save it as `10-IF_Statement.jpg` or `.png`.



```
do

# [TASK 10]
  file_last_modified=$(date -r "$file" + %s)
  if (($file_last_modified > $yesterdayTS));
  then

  # [TASK 11]
toBackup +=("$file")




```



```
cd # <-

# [TASK 8]
yesterdayTS=

declare -a toBackup

for file in $() [TASK 9]
do
  # [TASK 10]
  if (())
  then
    # [TASK 11]
  fi
done

# [TASK 12]

# [TASK 13]

# Congratulations! You completed
the final project for this course!

```



Task 11

1. In the `if-then` statement, add the `$file` that was updated in the past 24-hours to the `toBackup` array.
2. Since much of this wasn' t covered in the course, you may copy the code below and place after the `then` statement for this task:

```
toBackup+=($file)
```

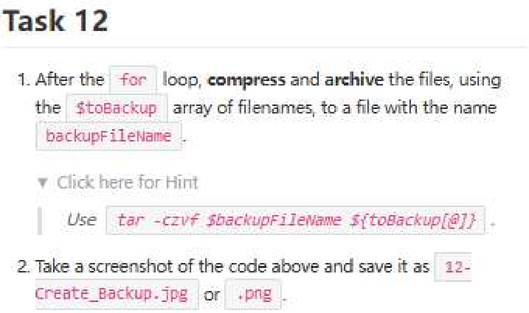
3. Take a screenshot of the code above and save it as `11-Add_File.jpg` or `.png`.

```
 if (($file_last_modified > $yesterdayTS));
 then

 # [TASK 11]
toBackup +=("$file")
```

```
# [TASK 12]

# [TASK 13]

# Congratulations! You completed the final project for this course!
```

Problems
theia@theia-professorjef: /home/project
theia@theia-professorjef: /home/project/newfile

```
theia@theia-professorjef:/home/project/new
theia@theia-professorjef:/home/project/new
theia@theia-professorjef:/home/project/new
```



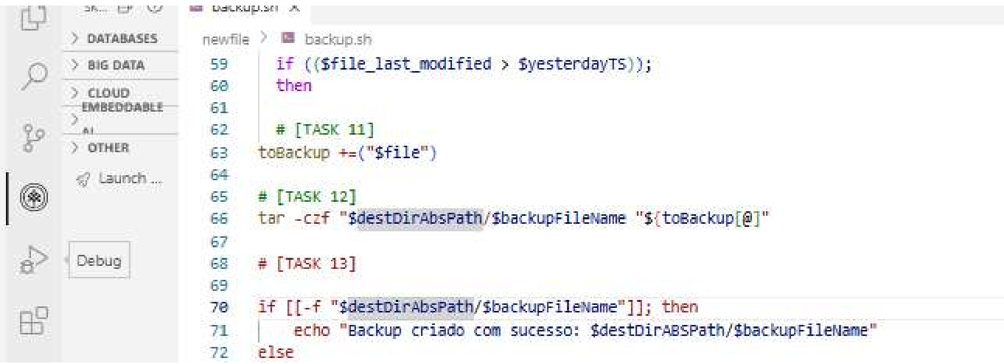
Task 12

1. After the `for` loop, **compress** and **archive** the files, using the `$toBackup` array of filenames, to a file with the name `backupFileName`.

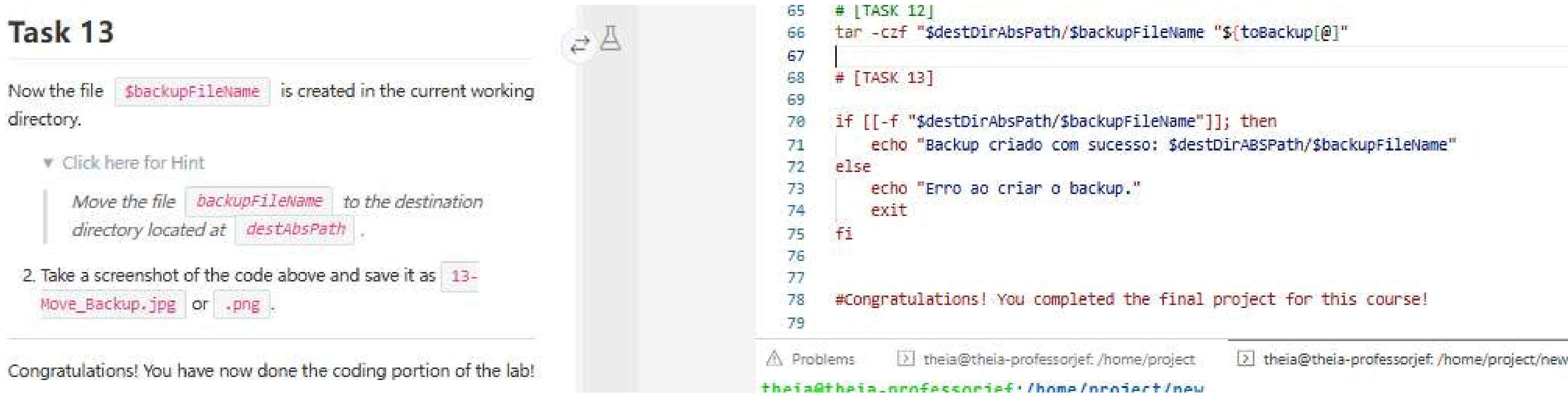
   ▼ Click here for Hint

   > *Use* `tar -czvf $backupFileName ${toBackup[@]}`.

2. Take a screenshot of the code above and save it as `12-Create_Backup.jpg` or `.png`.



```
  if (($file_last_modified > $yesterdayTS));
  then

  # [TASK 11]
toBackup +=("$file")

# [TASK 12]
tar -czf "$destDirAbsPath/$backupFileName "${toBackup[@]"

# [TASK 13]

if [[-f "$destDirAbsPath/$backupFileName"]]; then
    echo "Backup criado com sucesso: $destDirABSPath/$backupFileName"
else
```



Task 13

Now the file `$backupFileName` is created in the current working directory.

▾ Click here for Hint

> *Move the file `backupFileName` to the destination directory located at `destAbsPath` .*

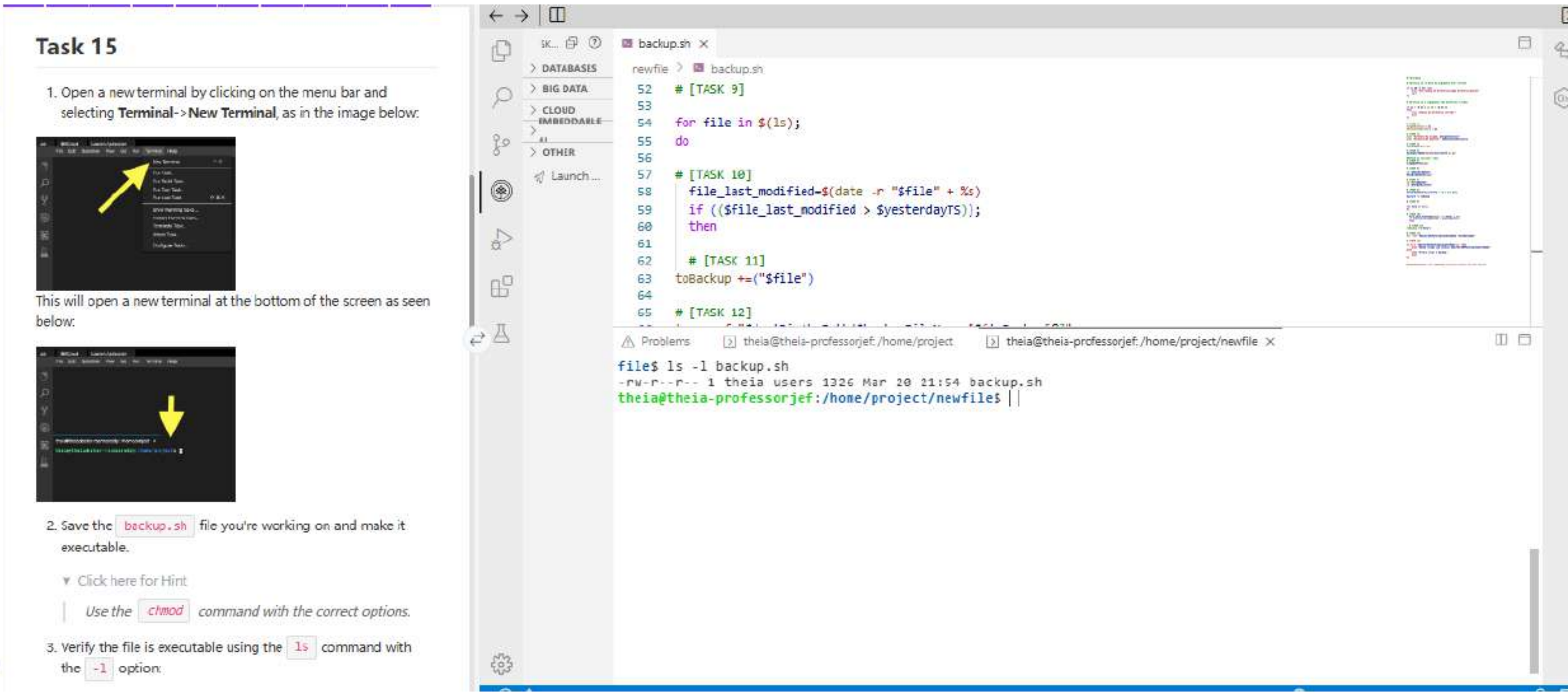
2. Take a screenshot of the code above and save it as `13-Move_Backup.jpg` or `.png`.

Congratulations! You have now done the coding portion of the lab!

```
# [TASK 12]
tar -czf "$destDirAbsPath/$backupFileName "${toBackup[@]"

# [TASK 13]

if [[-f "$destDirAbsPath/$backupFileName"]]; then
    echo "Backup criado com sucesso: $destDirABSPath/$backupFileName"
else
    echo "Erro ao criar o backup."
    exit
fi


#Congratulations! You completed the final project for this course!

```

Problems
theia@theia-professorjef: /home/project
theia@theia-professorjef: /home/project/new

```
theia@theia-professorjef:/home/project/new
```



Task 15

1. Open a new terminal by clicking on the menu bar and selecting **Terminal**->**New Terminal**, as in the image below:

This will open a new terminal at the bottom of the screen as seen below:

2. Save the `backup.sh` file you're working on and make it executable.

▼ Click here for Hint

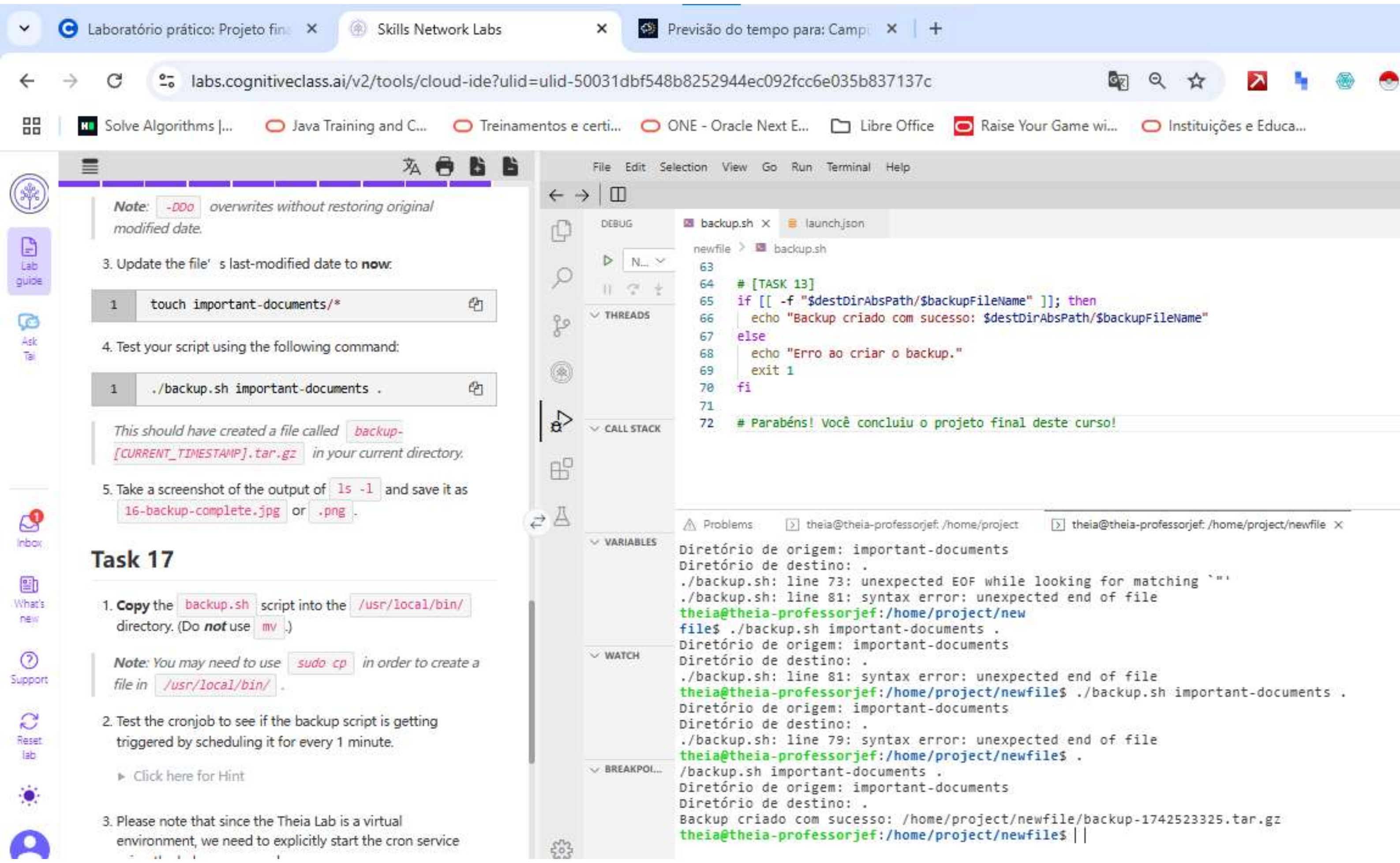
> *Use the `chmod` command with the correct options.*

3. Verify the file is executable using the `ls` command with the `-l` option:

DATABASES
BIG DATA
CLOUD
EMBEDDABLE
OTHER
Launch ...
backup.sh
newfile > backup.sh
Problems
theia@theia-professorjef: /home/project
theia@theia-professorjef: /home/project/newfile

```
# [TASK 9]

for file in $(ls);
do

# [TASK 10]
  file_last_modified-$(date -r "$file" + %s)
  if (($file_last_modified > $yesterdayTS));
  then

  # [TASK 11]
toBackup +=("$file")

# [TASK 12]
```

```
file$ ls -l backup.sh
-rw-r--r-- 1 theia users 1326 Mar 20 21:54 backup.sh
theia@theia-professorjef:/home/project/newfile$ |
```



Note: `-DDo` *overwrites without restoring original modified date.*

3. Update the file' s last-modified date to **now**:

```
touch important-documents/*
```

4. Test your script using the following command:

```
./backup.sh important-documents .
```

This should have created a file called `backup-[CURRENT_TIMESTAMP].tar.gz` *in your current directory.*

5. Take a screenshot of the output of `ls -l` and save it as `16-backup-complete.jpg` or `.png`.

Task 17

1. **Copy** the `backup.sh` script into the `/usr/local/bin/` directory. (Do ***not*** use `mv`.)

Note: *You may need to use* `sudo cp` *in order to create a file in* `/usr/local/bin/`.

2. Test the cronjob to see if the backup script is getting triggered by scheduling it for every 1 minute.

▸ Click here for Hint

3. Please note that since the Theia Lab is a virtual environment, we need to explicitly start the cron service

```

# [TASK 13]
if [[ -f "$destDirAbsPath/$backupFileName" ]]; then
  echo "Backup criado com sucesso: $destDirAbsPath/$backupFileName"
else
  echo "Erro ao criar o backup."
  exit 1
fi

# Parabéns! Você concluiu o projeto final deste curso!
```

```
Diretório de origem: important-documents
Diretório de destino: .
./backup.sh: line 73: unexpected EOF while looking for matching `"'
./backup.sh: line 81: syntax error: unexpected end of file
theia@theia-professorjef:/home/project/new
file$ ./backup.sh important-documents .
Diretório de origem: important-documents
Diretório de destino: .
./backup.sh: line 81: syntax error: unexpected end of file
theia@theia-professorjef:/home/project/newfile$ ./backup.sh important-documents .
Diretório de origem: important-documents
Diretório de destino: .
./backup.sh: line 79: syntax error: unexpected end of file
theia@theia-professorjef:/home/project/newfile$ .
/backup.sh important-documents .
Diretório de origem: important-documents
Diretório de destino: .
Backup criado com sucesso: /home/project/newfile/backup-1742523325.tar.gz
theia@theia-professorjef:/home/project/newfile$ ||
```