

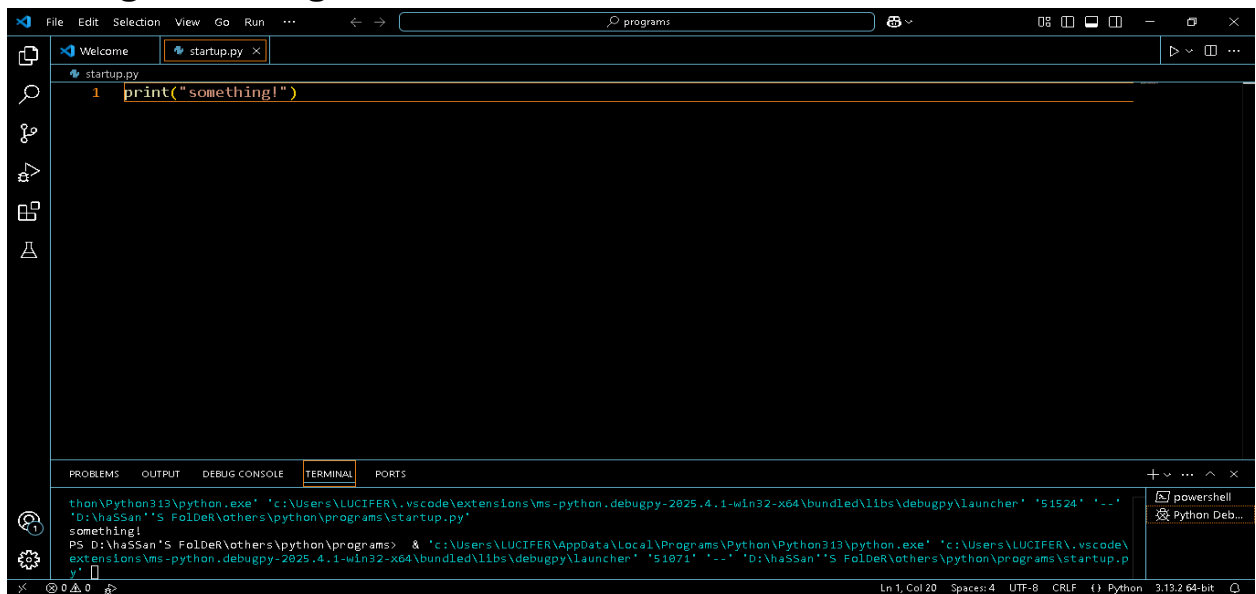
**Name: Hassan Nawaz**

**Username: HassanNawazBhatti**

**Roll no: BCS23042**

---

- **Printing something**



```
File Edit Selection View Go Run ... programs
Welcome startup.py
1 print("something!")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
thon\Python313\python.exe 'c:\Users\LUCIFER\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundle\libs\debugpy\launcher' '51524' '--'
'D:\haSSan'S FOLDER\others\python\programs\startup.py'
something!
PS D:\haSSan'S FOLDER\others\python\programs> & 'c:\Users\LUCIFER\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\LUCIFER\.vscode\
extensions\ms-python.debugpy-2025.4.1-win32-x64\bundle\libs\debugpy\launcher' '51071' '--' 'D:\haSSan'S FOLDER\others\python\programs\startup.p
y'
```


- **Commenting a statement:**



```
File Edit Selection View Go Run ... programs
Welcome startup.py
1 print("something!")
2 # this will not be executed

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\haSSan'S FOLDER\others\python\programs>
PS D:\haSSan'S FOLDER\others\python\programs> cd 'd:\haSSan'S FOLDER\others\python\programs' & 'c:\Users\LUCIFER\AppData\Local\Programs\py
thon\Python313\python.exe' 'c:\Users\LUCIFER\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundle\libs\debugpy\launcher' '51613' '--'
'D:\haSSan'S FOLDER\others\python\programs\startup.py'
something!
PS D:\haSSan'S FOLDER\others\python\programs>
```

- **Multiple statements in a line:**



The screenshot shows the VS Code editor with a file named `startup.py` open. The code in the editor is:

```
1 #printing 2 statements in a line
2 print("statement1") ; print("statement2")
3
```

The bottom panel shows the TERMINAL output:

```
PS D:\haSSan\S_FolDeR\others\python\programs> .d;; cd 'd:\haSSan\S_FolDeR\others\python\programs'; & 'c:\Users\LUCIFER\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\LUCIFER\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundle\libs\debugpy\launcher' '51649' '-'
statement1
statement2
PS D:\haSSan\S_FolDeR\others\python\programs>
```

The status bar at the bottom indicates the cursor is at Line 3, Column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.13.2 64-bit.

- **Tab and Space Identations:**



The screenshot shows the VS Code editor with a file named `startup.py` open. The code in the editor is:

```
1 x=1
2 if x>0:
3     print("this is single space indentation")
4     print("this is also single space indentation")
```

The bottom panel shows the TERMINAL output:

```
PS D:\haSSan\S_FolDeR\others\python\programs> .d;; cd 'd:\haSSan\S_FolDeR\others\python\programs'; & 'c:\Users\LUCIFER\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\LUCIFER\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundle\libs\debugpy\launcher' '51670' '-'
this is single space indentation
this is also single space indentation
PS D:\haSSan\S_FolDeR\others\python\programs>
```

The status bar at the bottom indicates the cursor is at Line 4, Column 47, with 4 spaces, UTF-8 encoding, CRLF line endings, and Python 3.13.2 64-bit.

The screenshot shows the Visual Studio Code editor with a file named `startup.py` open. The code in the editor is as follows:

```
1 x=1
2 if x>0:
3     print("this is single tab indentation")
4     print("this is also single tab indentation")
```

The terminal at the bottom shows the command used to run the script and its output:

```
PS D:\haSSan'S FOLDeR\others\python\programs> d:; cd 'd:\haSSan'S FOLDeR\others\python\programs'; & 'c:\Users\LUCIFER\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\LUCIFER\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundle\libs\debugpy\launcher' '51714' '--' 'D:\haSSan'S FOLDeR\others\python\programs\startup.py'
this is single tab indentation
this is also single tab indentation
PS D:\haSSan'S FOLDeR\others\python\programs>
```

The screenshot shows the Visual Studio Code editor with a file named `startup.py` open. The code in the editor is as follows:

```
1 x=1
2 if x>0:
3     print("this is single space + tab indentation")
4     print("this is also single space + tab indentation")
```

The terminal at the bottom shows the command used to run the script and its output:

```
PS D:\haSSan'S FOLDeR\others\python\programs> d:; cd 'd:\haSSan'S FOLDeR\others\python\programs'; & 'c:\Users\LUCIFER\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\LUCIFER\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundle\libs\debugpy\launcher' '51729' '--' 'D:\haSSan'S FOLDeR\others\python\programs\startup.py'
this is single space + tab indentation
this is also single space + tab indentation
PS D:\haSSan'S FOLDeR\others\python\programs>
```

- **Data types:**



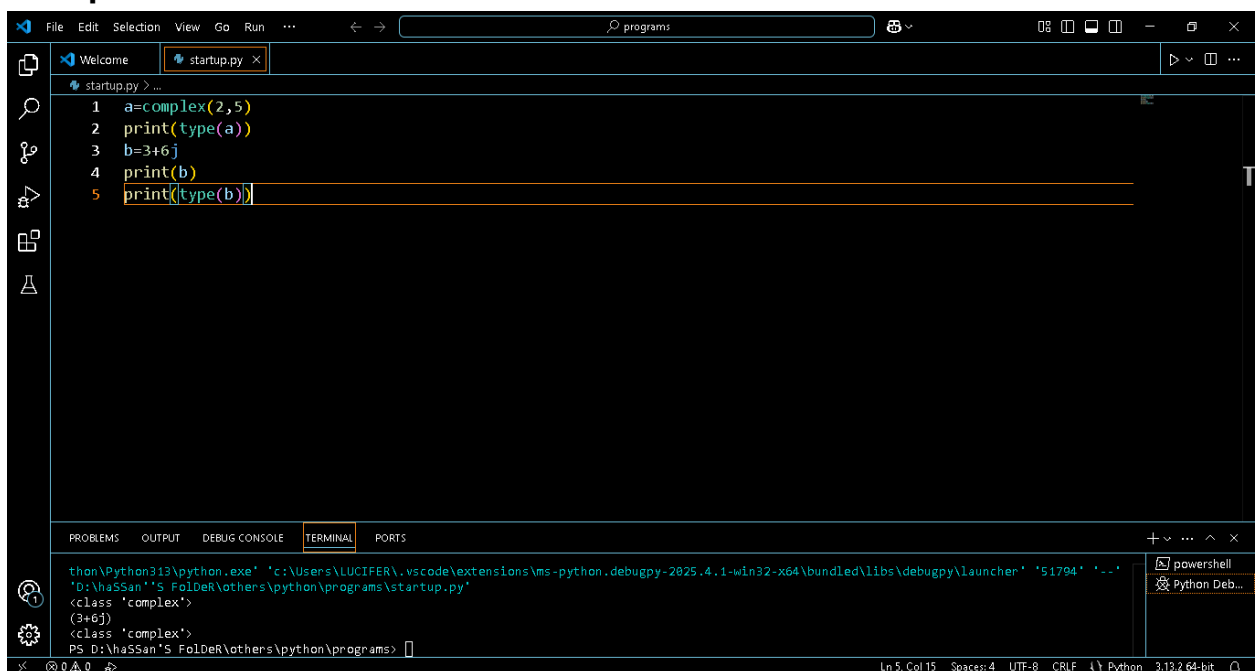
The screenshot shows the VS Code editor with a file named 'startup.py' open. The code defines five variables: 'a' (integer), 'b' (float), 'c' (string), 'd' (boolean), and 'e' (boolean). Each variable is printed along with its type using the `print(type(variable))` function. The terminal at the bottom shows the output of the script, displaying the class names for each variable: `<class 'int'>`, `<class 'float'>`, `<class 'str'>`, `<class 'bool'>`, and `<class 'bool'>`. The status bar at the bottom indicates the file is at line 10, column 14, using UTF-8 encoding with CRLF line endings.

```
1 a=123
2 print(type(a))
3 b=1.23
4 print(type(b))
5 c="hello"
6 print(type(c))
7 d=True
8 print(type(d))
9 e=False
10 print(type(e))
```

Terminal Output:

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
<class 'bool'>
PS D:\haSSan'S FOLDER\others\python\programs>
```

- **Complex function:**



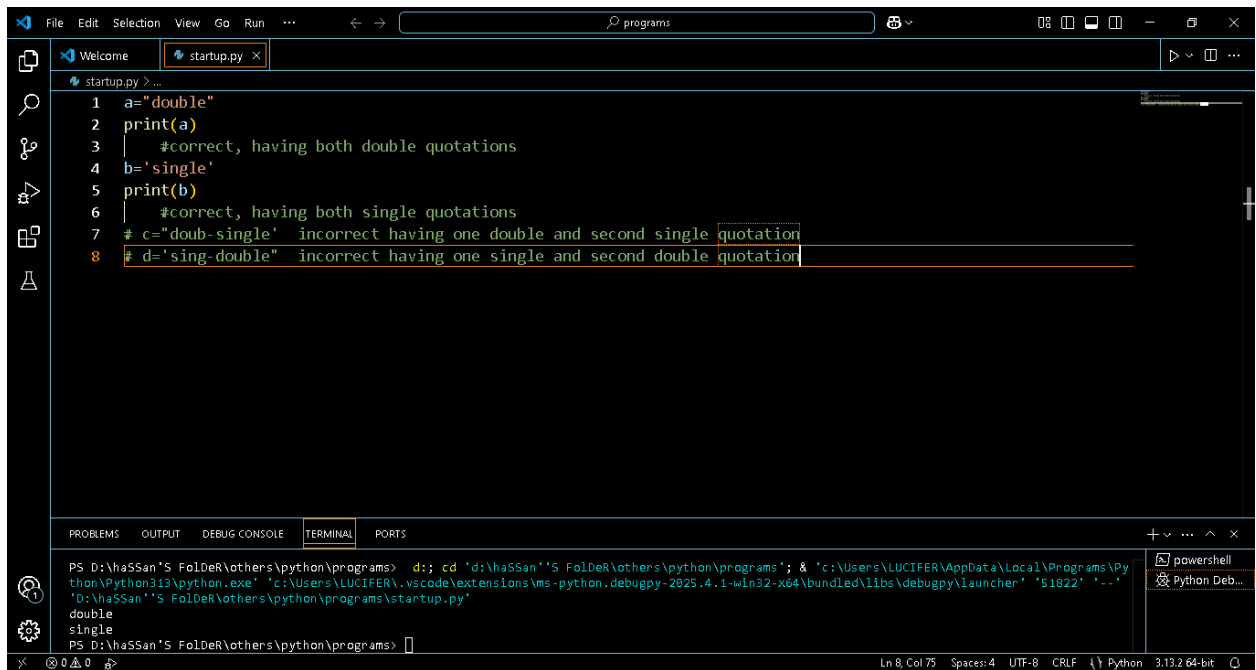
The screenshot shows the VS Code editor with a file named 'startup.py' open. The code defines two variables: 'a' using the `complex()` function with arguments 2 and 5, and 'b' as a complex number `3+6j`. Both variables are printed along with their types. The terminal at the bottom shows the output of the script, displaying the class names for each variable: `<class 'complex'>` for both 'a' and 'b'. The status bar at the bottom indicates the file is at line 5, column 15, using UTF-8 encoding with CRLF line endings.

```
1 a=complex(2,5)
2 print(type(a))
3 b=3+6j
4 print(b)
5 print(type(b))
```

Terminal Output:

```
then\Python313\python.exe" "c:\Users\LUCIFER\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundle\libs\debugpy\launcher" "51794" "...
D:\haSSan'S FOLDER\others\python\programs\startup.py"
<class 'complex'>
(3+6j)
<class 'complex'>
PS D:\haSSan'S FOLDER\others\python\programs>
```

- String using single and double quotes:

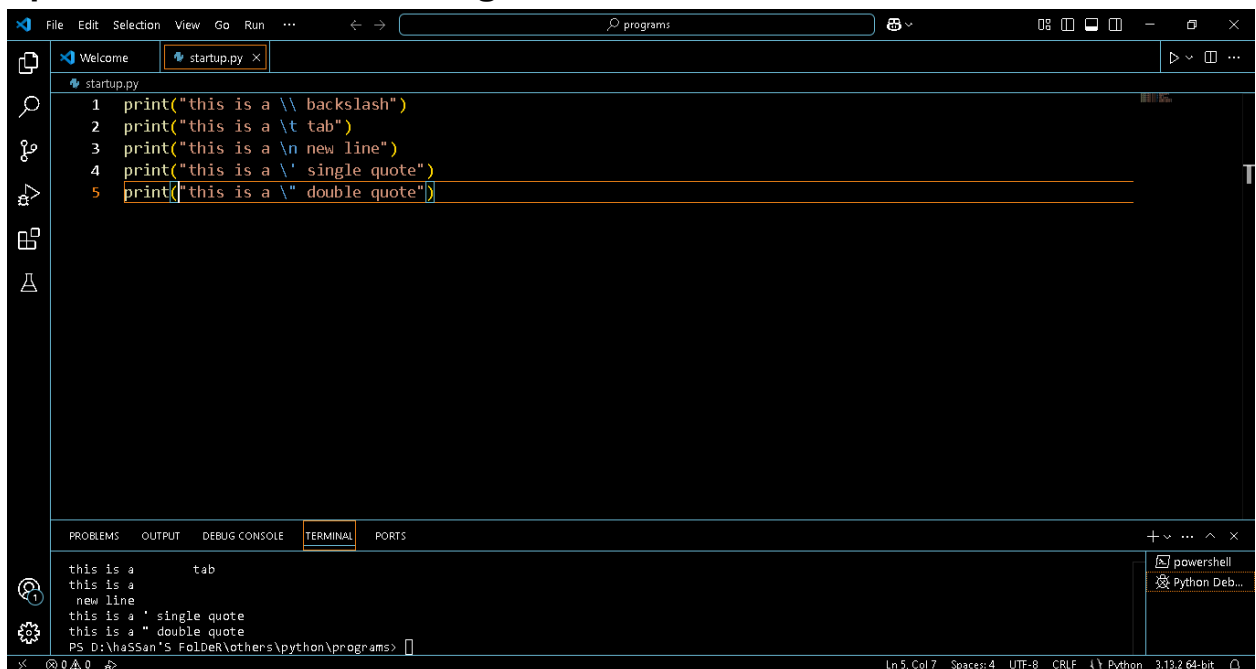


```
1 a="double"
2 print(a)
3 #correct, having both double quotations
4 b='single'
5 print(b)
6 #correct, having both single quotations
7 # c="doub-single" incorrect having one double and second single quotation
8 # d='sing-double' incorrect having one single and second double quotation
```

Terminal output:

```
PS D:\haSSan'S FoLDeR\others\python\programs> d:; cd 'd:\haSSan'S FoLDeR\others\python\programs'; & 'c:\Users\LUCIFER\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\LUCIFER\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundled\libs\debugpy\launcher' '51822' '--' 'D:\haSSan'S FoLDeR\others\python\programs\startup.py'
double
single
PS D:\haSSan'S FoLDeR\others\python\programs>
```

- Special characters in a string:

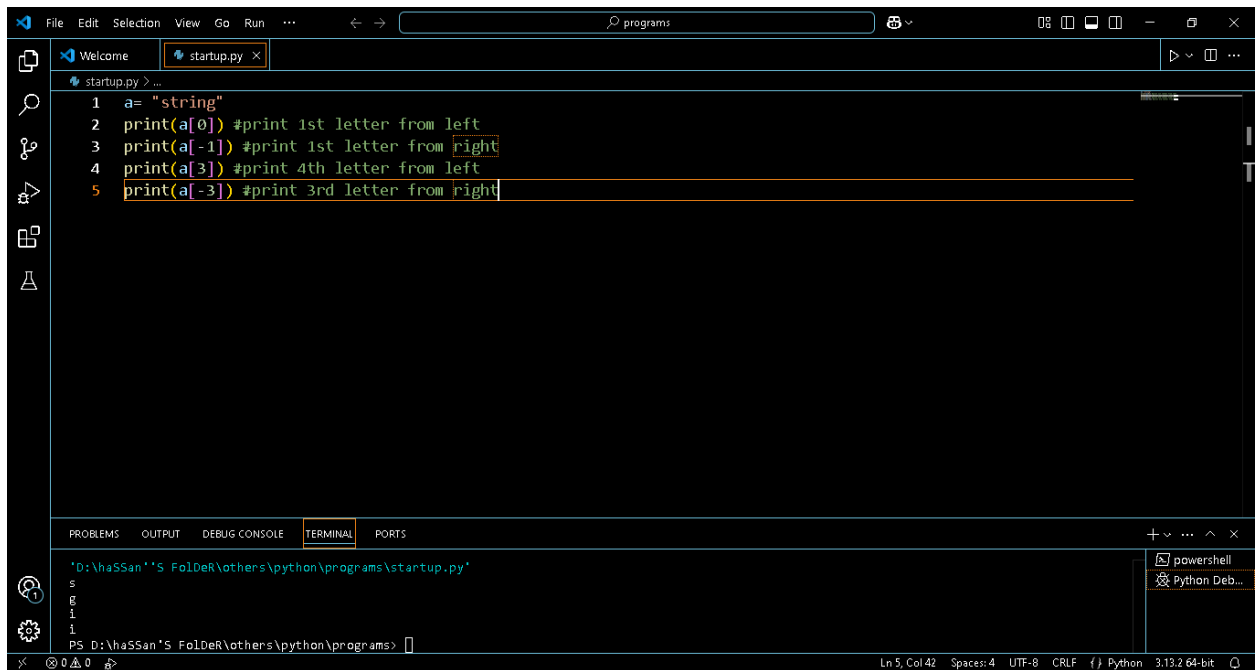


```
1 print("this is a \\ backslash")
2 print("this is a \t tab")
3 print("this is a \n new line")
4 print("this is a \' single quote")
5 print("this is a \" double quote")
```

Terminal output:

```
this is a      tab
this is a
new line
this is a ' single quote
this is a " double quote
PS D:\haSSan'S FoLDeR\others\python\programs>
```

- String indices:

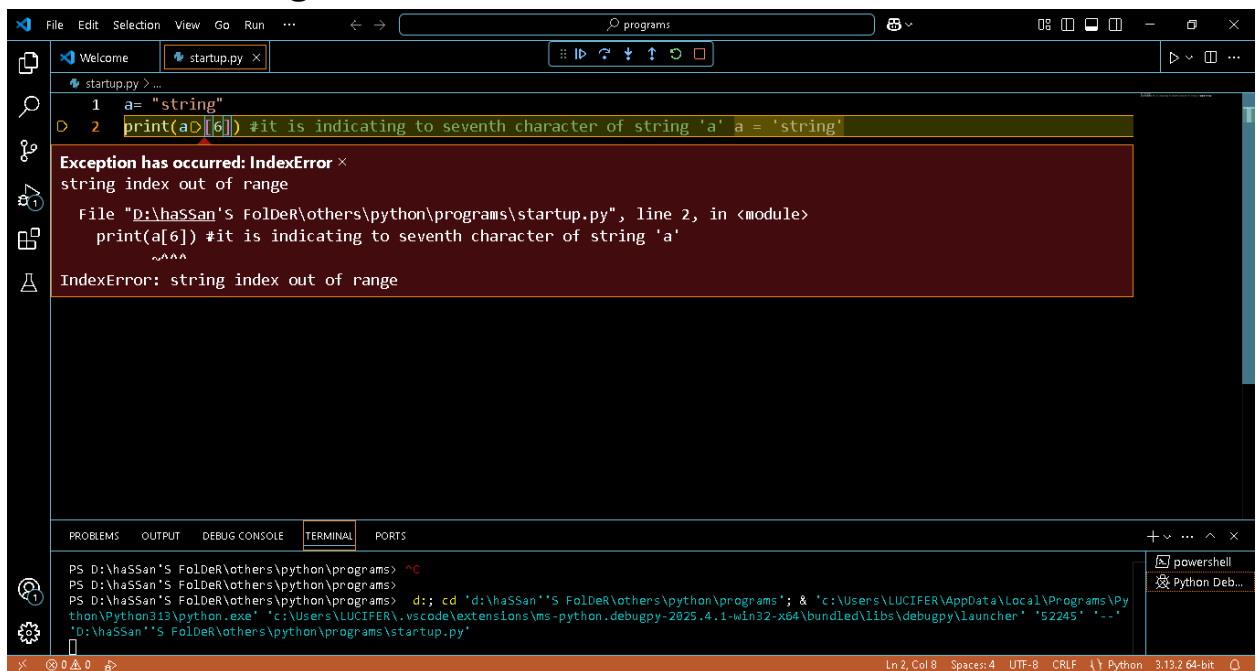


The screenshot shows the Visual Studio Code editor with a file named `startup.py` open. The code contains five lines of Python code that demonstrate string indexing on the string `"string"`. The first line assigns the string to variable `a`. The subsequent lines use `print` statements to access characters at various indices, with comments explaining the direction of access (left or right). The interface includes a sidebar with icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The bottom panel shows the TERMINAL tab with the command prompt path `PS D:\haSSan\S_FolDeR\others\python\programs>`.

```
1 a= "string"
2 print(a[0]) #print 1st letter from left
3 print(a[-1]) #print 1st letter from right
4 print(a[3]) #print 4th letter from left
5 print(a[-3]) #print 3rd letter from right
```

TERMINAL: `PS D:\haSSan\S_FolDeR\others\python\programs>`

- Error in accessing element:



The screenshot shows the Visual Studio Code editor with the same `startup.py` file. The second line of code, `print(a[6])`, is highlighted, and a red arrow points to the index `6`. A large red error message box is displayed over the code, indicating an `IndexError` with the message "string index out of range". The error details specify the file path and line number. The bottom panel shows the TERMINAL tab with the command prompt path `PS D:\haSSan\S_FolDeR\others\python\programs>`.

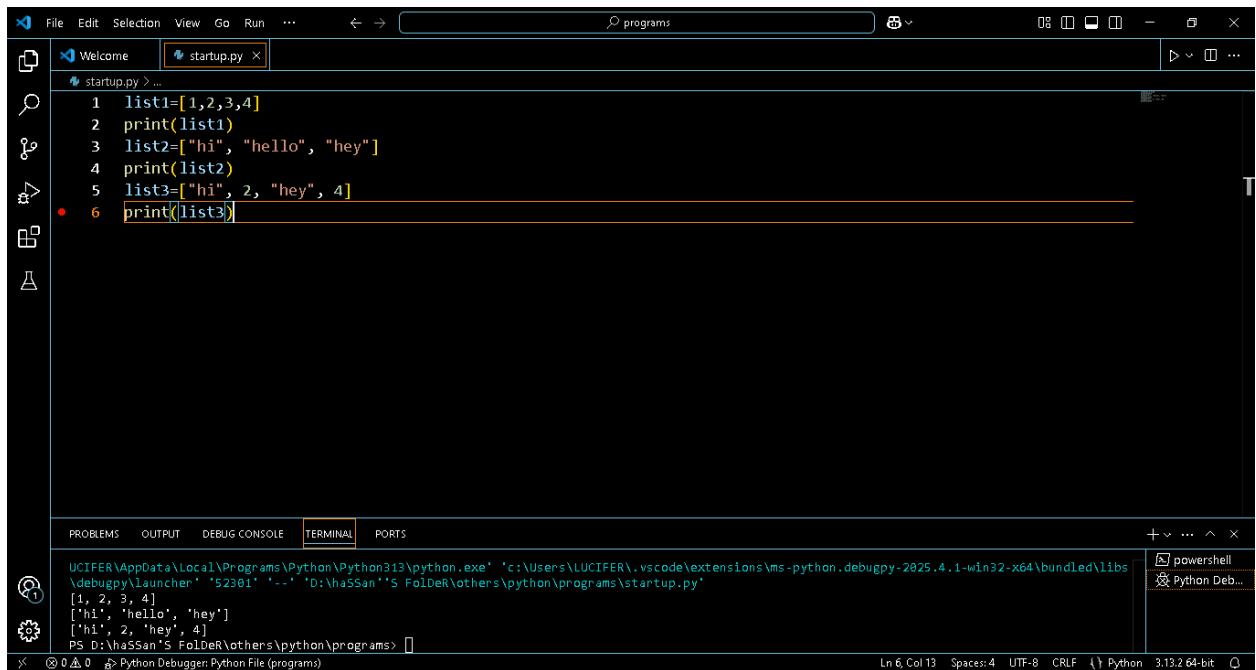
```
1 a= "string"
2 print(a[6]) #it is indicating to seventh character of string 'a' a = 'string'
```

**Exception has occurred: IndexError**  
string index out of range

File "D:\haSSan\S\_FolDeR\others\python\programs\startup.py", line 2, in <module>  
print(a[6]) #it is indicating to seventh character of string 'a'  
~~~~~  
IndexError: string index out of range

TERMINAL: `PS D:\haSSan\S_FolDeR\others\python\programs>`

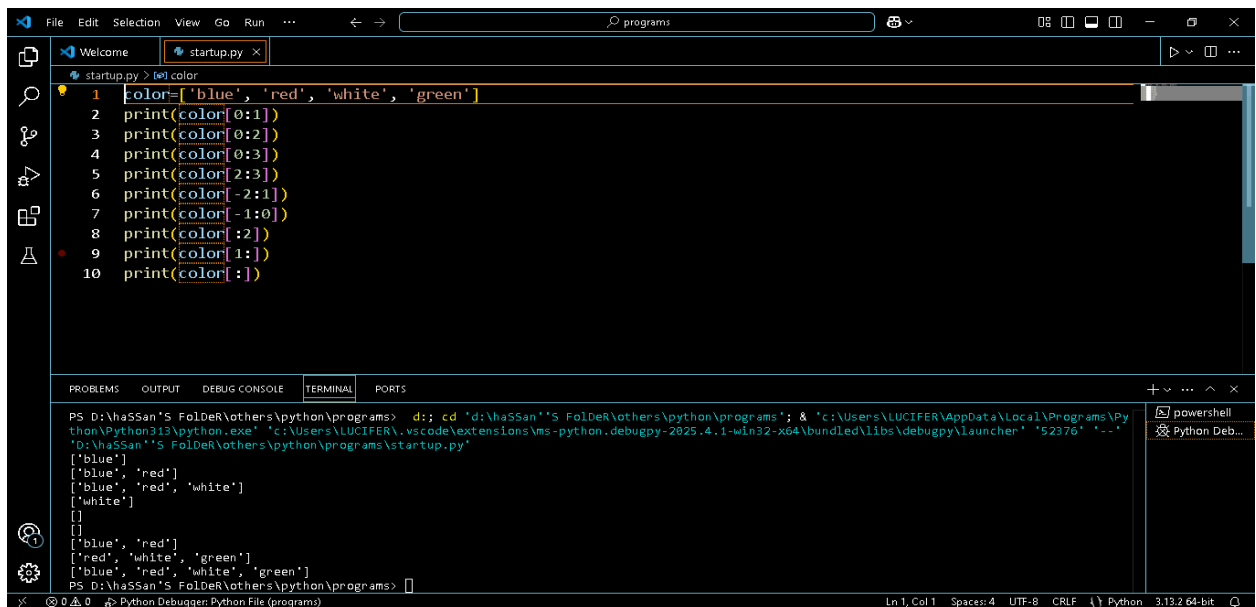
- **Lists:**



```
File Edit Selection View Go Run ... programs
Welcome startup.py x
1 list1=[1,2,3,4]
2 print(list1)
3 list2=["hi", "hello", "hey"]
4 print(list2)
5 list3=["hi", 2, "hey", 4]
6 print(list3)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
UCIFER\AppData\Local\Programs\Python\Python313\python.exe "c:\Users\LUCIFER\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundle...
[1, 2, 3, 4]
['hi', 'hello', 'hey']
['hi', 2, 'hey', 4]
PS D:\haSSan'S FoLdeR\others\python\programs>
```

- **List slicing:**



```
File Edit Selection View Go Run ... programs
Welcome startup.py x
1 color=['blue', 'red', 'white', 'green']
2 print(color[0:1])
3 print(color[0:2])
4 print(color[0:3])
5 print(color[2:3])
6 print(color[-2:1])
7 print(color[-1:0])
8 print(color[:2])
9 print(color[1:])
10 print(color[:])

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\haSSan'S FoLdeR\others\python\programs> d;; cd 'd:\haSSan'S FoLdeR\others\python\programs'; & 'c:\Users\LUCIFER\AppData\Local\Programs\Py...
['blue']
['blue', 'red']
['blue', 'red', 'white']
['white']
[]
['blue', 'red']
['red', 'white', 'green']
['blue', 'red', 'white', 'green']
PS D:\haSSan'S FoLdeR\others\python\programs>
```

---

**-----THE END-----**

---