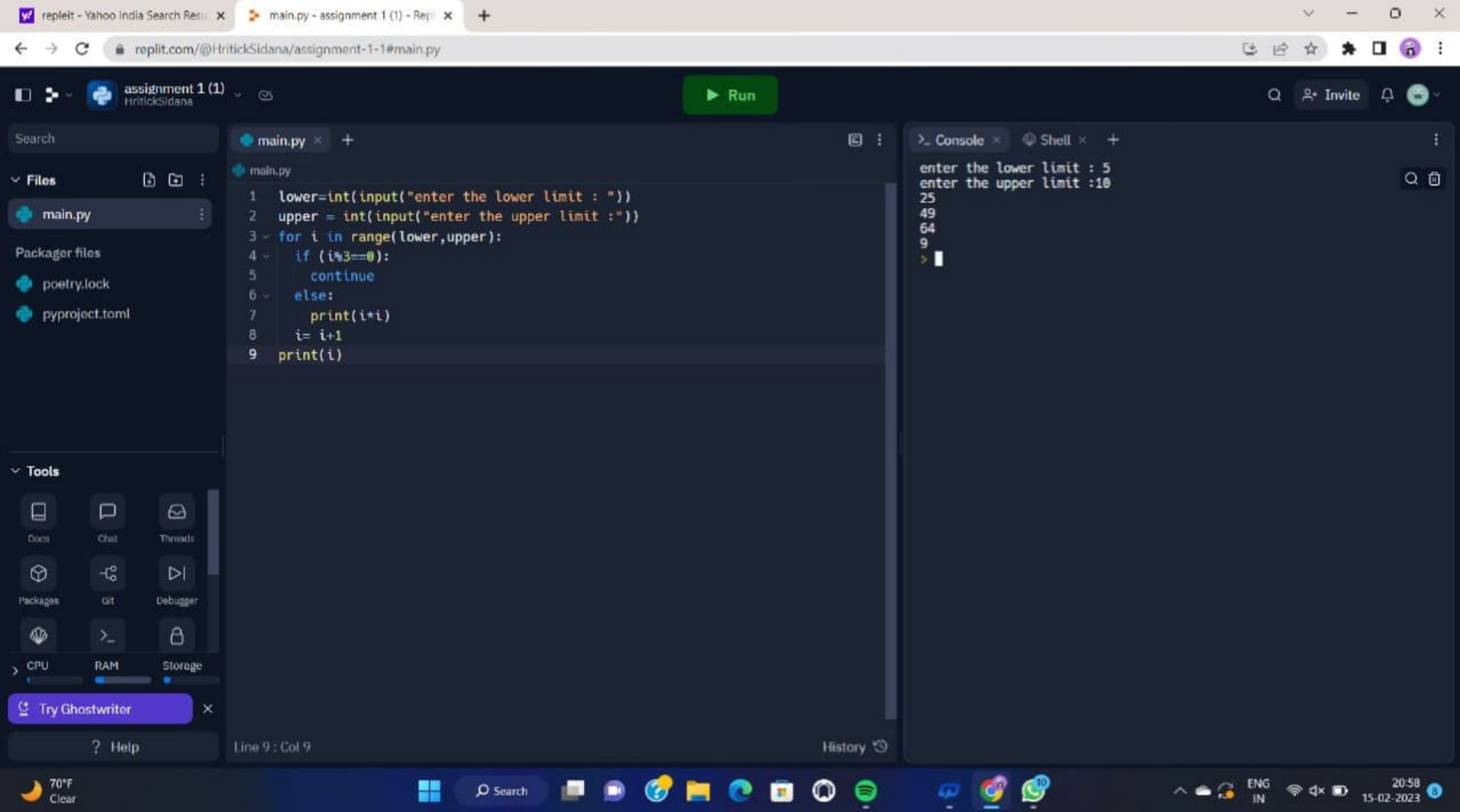
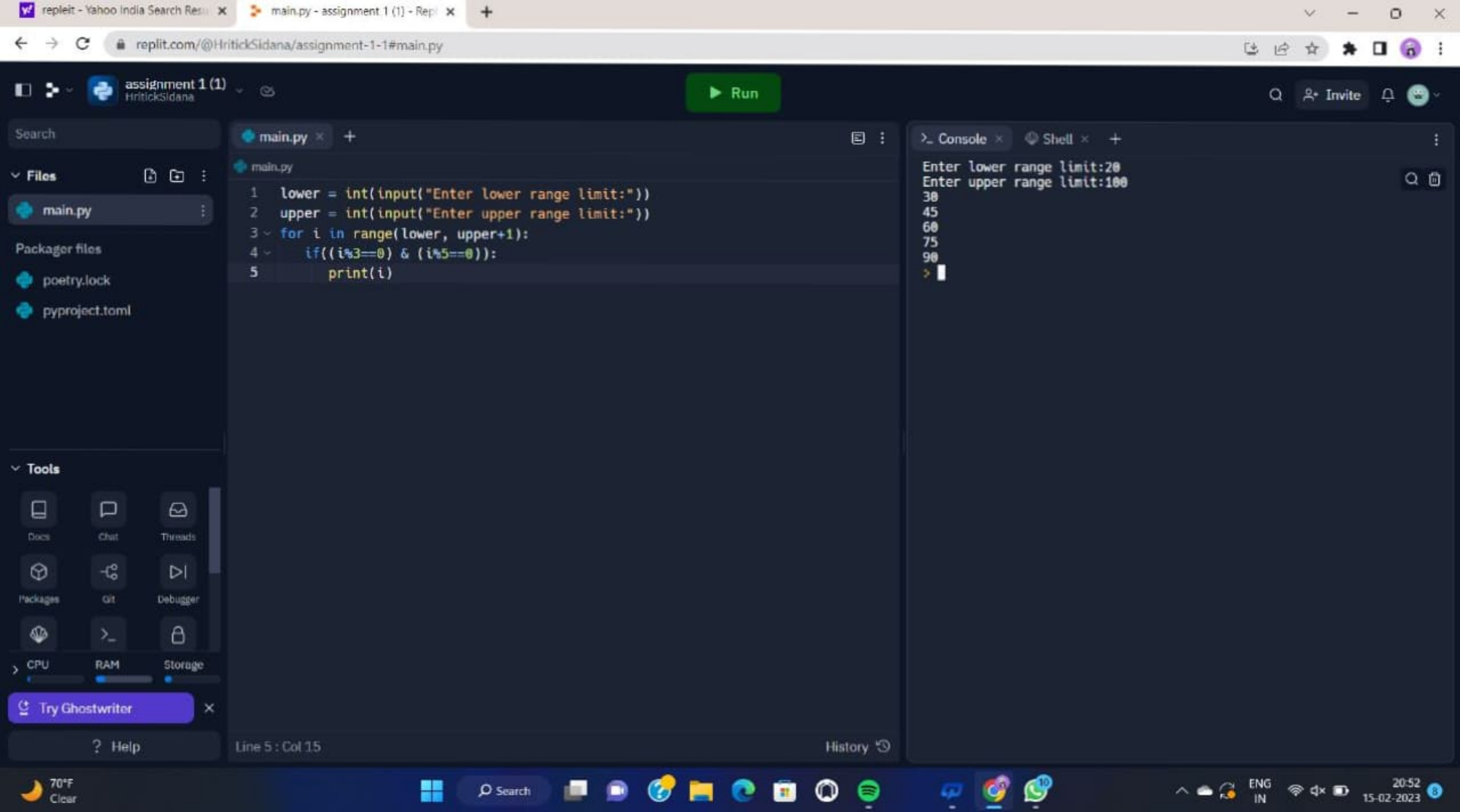
```
lst = []
n = int(input("Enter number of elements : "))
for i in range(0, n):
    ele = int(input())

    lst.append(ele) # adding the element

print(lst)
def sqrt_list(lst):
    ret = []
    for i in lst:
        ret.append(i**0.5 )
    return ret
print(sqrt_list(lst))
```



```
lower=int(input("enter the lower limit : "))
upper = int(input("enter the upper limit :"))
for i in range(lower,upper):
  if (i%3==0):
    continue
  else:
    print(i*i)
  i= i+1
print(i)
```

```
enter the lower limit : 5
enter the upper limit :10
25
49
64
9
>
```

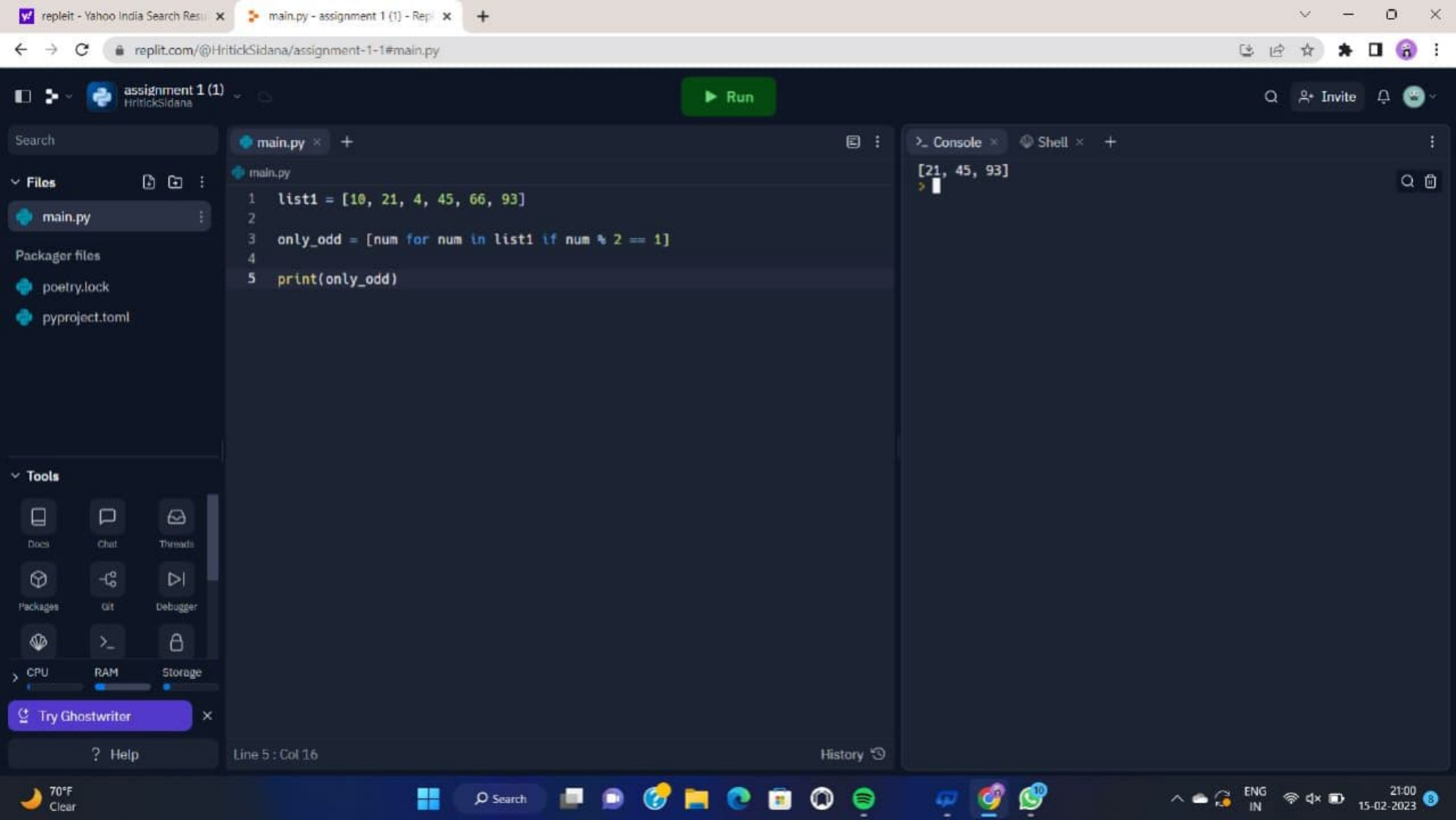
```
lower = int(input("Enter lower range limit:"))
upper = int(input("Enter upper range limit:"))
for i in range(lower, upper+1):
   if((i%3==0) & (i%5==0)):
      print(i)
```



```python
lower = int(input("Enter lower range limit:"))
upper = int(input("Enter upper range limit:"))
for i in range(lower, upper+1):
    if((i%3==0) & (i%5==0)):
        print(i)
```

```
Enter lower range limit:20
Enter upper range limit:100
30
45
60
75
90
>
```

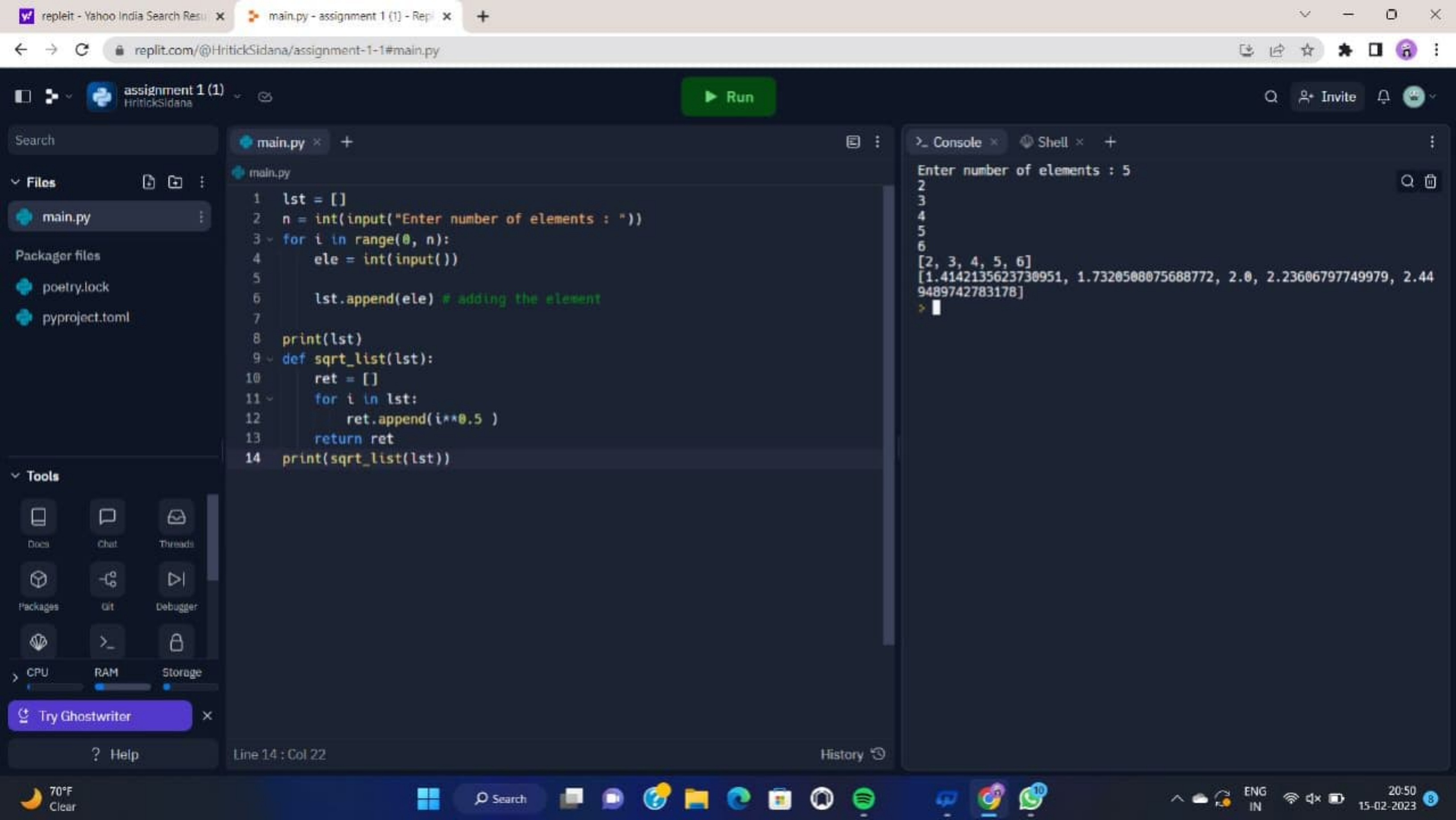
```
def vowelOrConsonant(x):

    if (x == 'a' or x == 'e' or
        x == 'i' or x == 'o' or x ==
'u'):
        print("Vowel")
    else:
        print("Consonant")

vowelOrConsonant('a')
vowelOrConsonant('f')
```



```python
list1 = [10, 21, 4, 45, 66, 93]

only_odd = [num for num in list1 if num % 2 == 1]

print(only_odd)
```

```
[21, 45, 93]
>
```

```
def shortestDistance(S, X):

    # Find distance from occurrences of X
    # appearing before current character.
    inf = float('inf')
    prev = inf
    ans = []
    for i,j in enumerate(S):
        if S[i] == X:
            prev = i
        if (prev == inf) :
            ans.append(inf)
        else :
            ans.append(i - prev)


    # Find distance from occurrences of X
    # appearing after current character and
    # compare this distance with earlier.
    prev = inf
    for i in range(len(S) - 1, -1, -1):
        if S[i] == X:
            prev = i
        if (X != inf):
            ans[i] = min(ans[i], prev - i)

    # return array of distance
    return ans


# Driver code
S = "geeksforgeeks"
X = "g"

# Function call to print answer
print(shortestDistance(S, X))
```

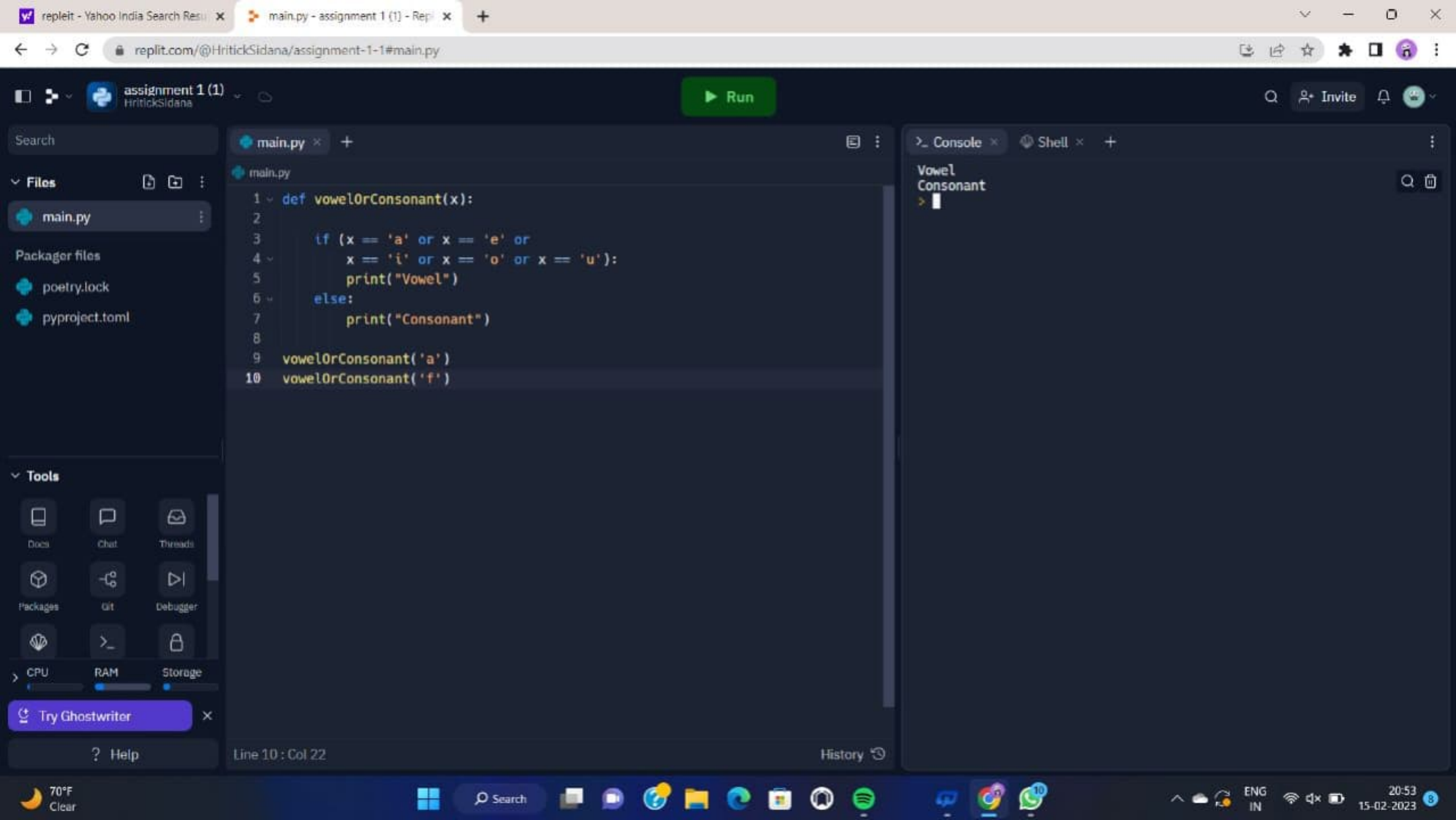


main.py

```python
lst = []
n = int(input("Enter number of elements : "))
for i in range(0, n):
    ele = int(input())

    lst.append(ele) # adding the element

print(lst)
def sqrt_list(lst):
    ret = []
    for i in lst:
        ret.append(i**0.5 )
    return ret
print(sqrt_list(lst))
```

```
Enter number of elements : 5
2
3
4
5
6
[2, 3, 4, 5, 6]
[1.4142135623730951, 1.7320508075688772, 2.0, 2.23606797749979, 2.44
9489742783178]
> 
```

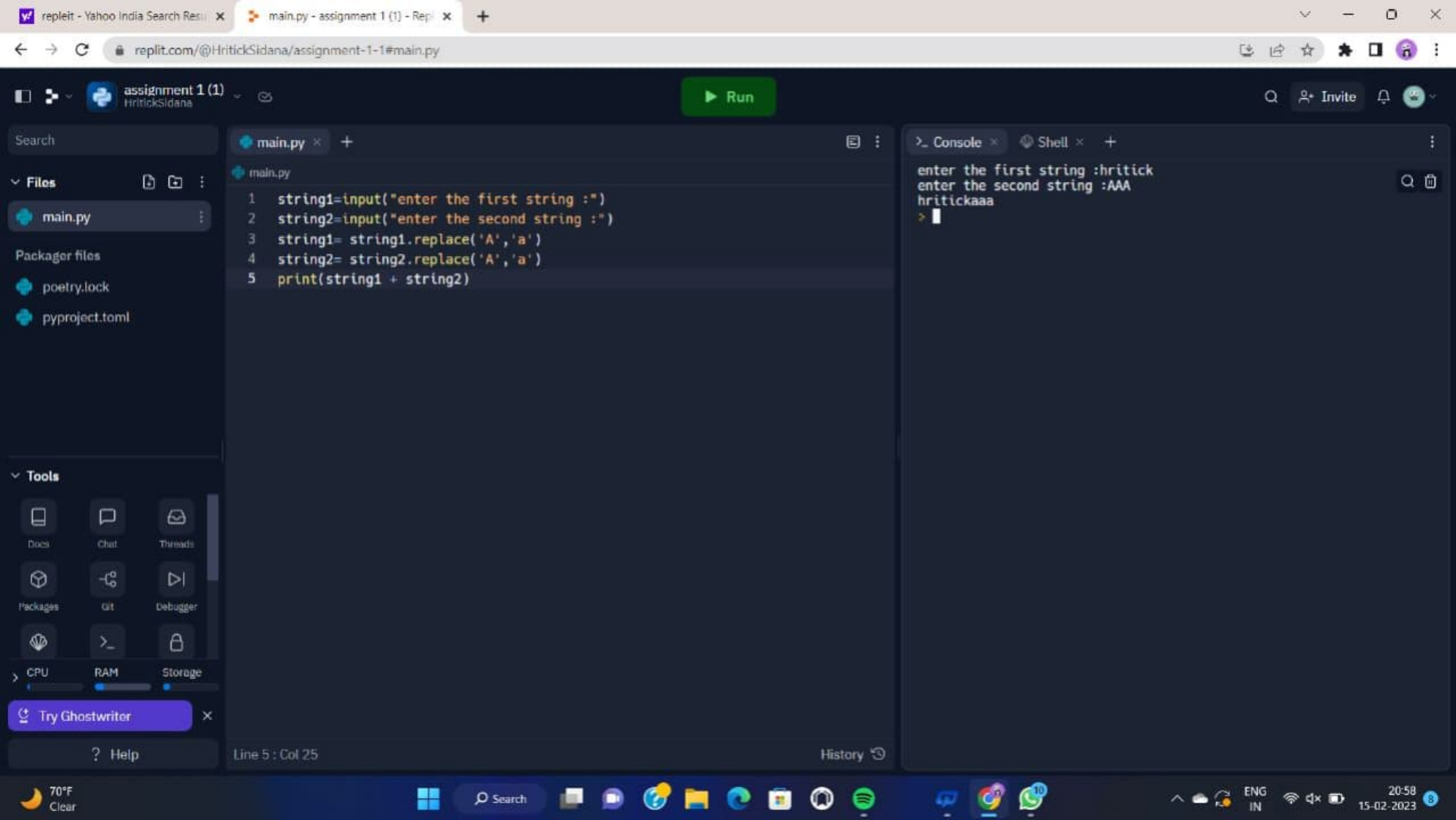
```
def Check_Vow(string, vowels):
    string = string.casefold()
    count = {}.fromkeys(vowels, 0)

    for character in string:
        if character in count:
            count[character] += 1
    return count
vowels = 'aeiou'
string = input("enter the string : ")
print (Check_Vow(string, vowels))
```



```
def vowelOrConsonant(x):

    if (x == 'a' or x == 'e' or
        x == 'i' or x == 'o' or x == 'u'):
        print("Vowel")
    else:
        print("Consonant")

vowelOrConsonant('a')
vowelOrConsonant('f')
```

```
Vowel
Consonant
>
```

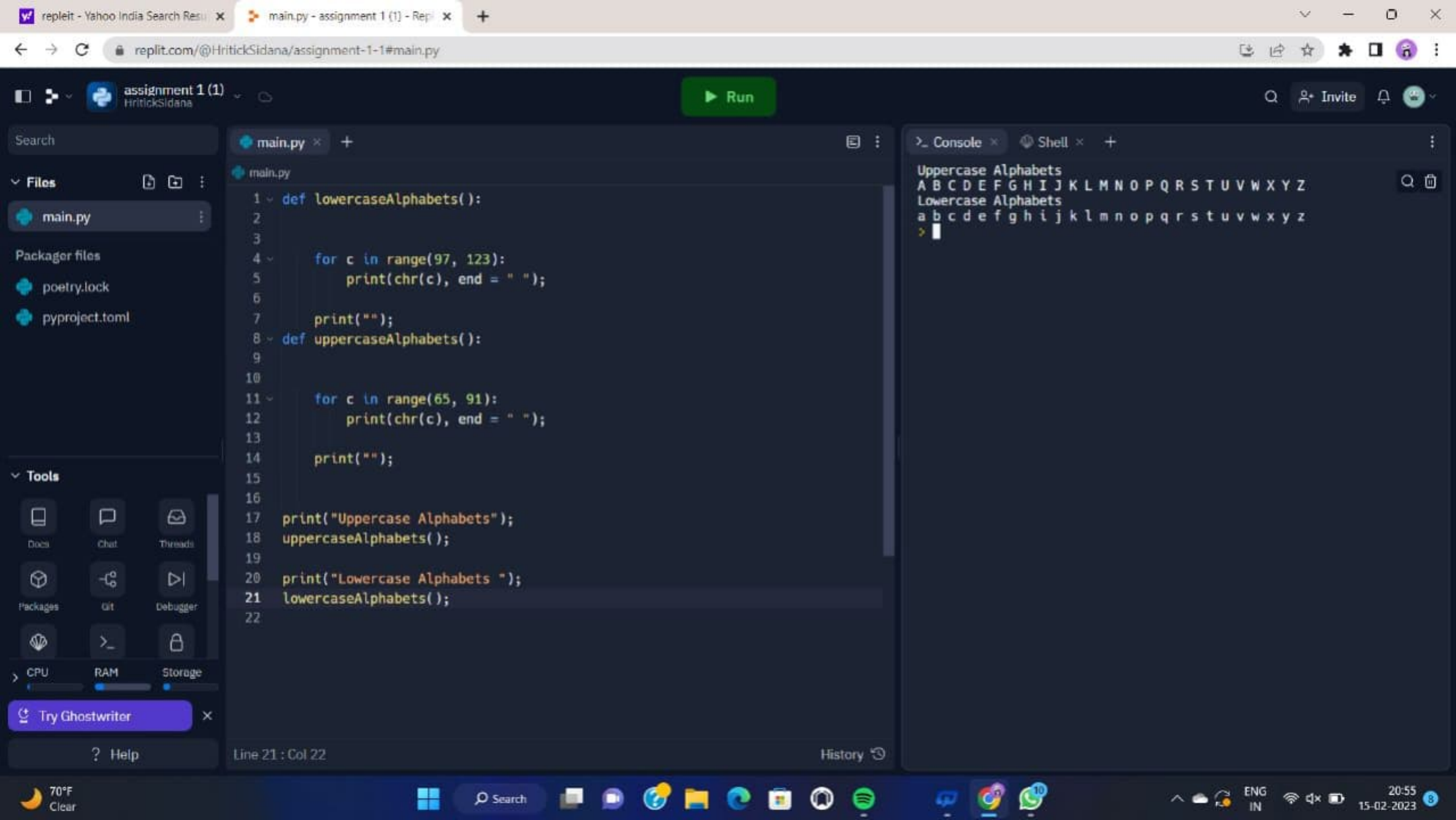
```
def lowercaseAlphabets():


    for c in range(97, 123):
        print(chr(c), end = " ");

    print("");
def uppercaseAlphabets():


    for c in range(65, 91):
        print(chr(c), end = " ");

    print("");


print("Uppercase Alphabets");
uppercaseAlphabets();

print("Lowercase Alphabets ");
lowercaseAlphabets();
```



```python
string1=input("enter the first string :")
string2=input("enter the second string :")
string1= string1.replace('A','a')
string2= string2.replace('A','a')
print(string1 + string2)
```

```
enter the first string :hritick
enter the second string :AAA
hritickaaa
>
```

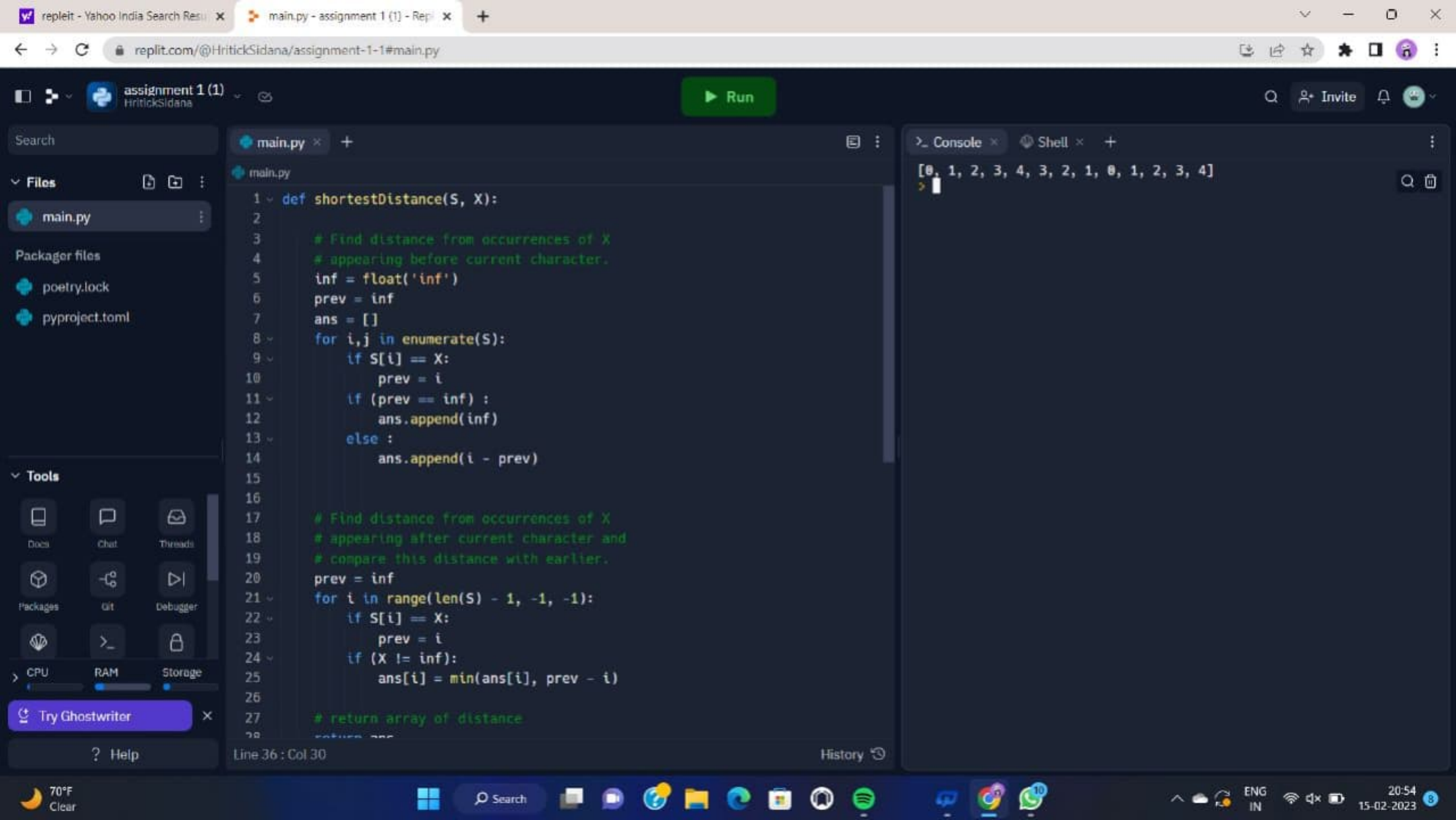
```
NumList = []
Even_Sum = 0
Number = int(input("Please enter the Total
Number of List Elements: "))
for i in range(1, Number + 1):
    value = int(input("Please enter the Value of
%d Element : " %i))
    NumList.append(value)

for j in range(Number):
    if(NumList[j] % 2 == 0):
        Even_Sum = Even_Sum + NumList[j]
print("\nThe Sum of Even Numbers in this List
=  ", Even_Sum)
```



```python
def lowercaseAlphabets():


    for c in range(97, 123):
        print(chr(c), end = " ");

    print("");
def uppercaseAlphabets():


    for c in range(65, 91):
        print(chr(c), end = " ");

    print("");


print("Uppercase Alphabets");
uppercaseAlphabets();

print("Lowercase Alphabets ");
lowercaseAlphabets();

```

```
Uppercase Alphabets
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Lowercase Alphabets
a b c d e f g h i j k l m n o p q r s t u v w x y z
>
```

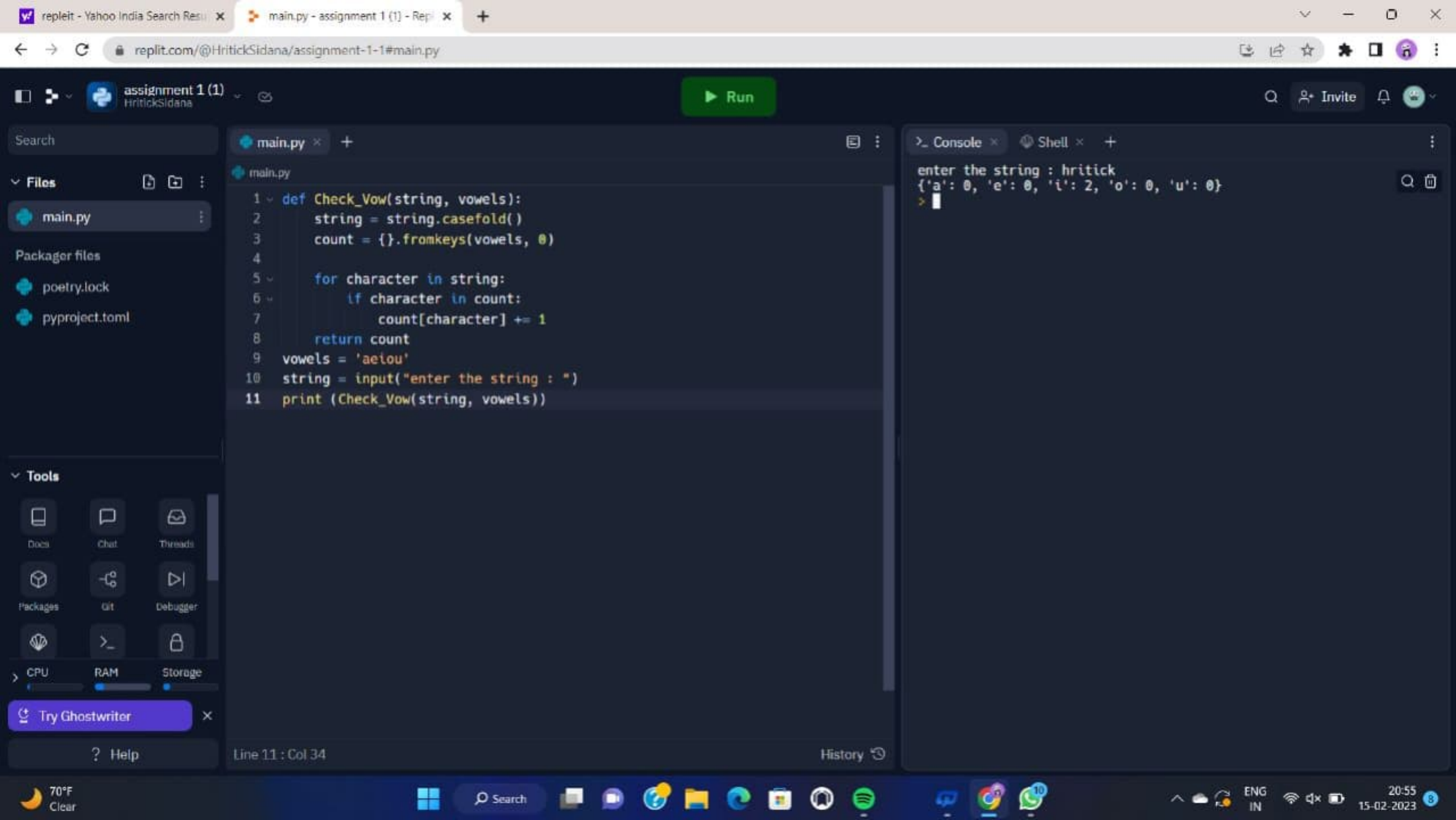
```
lower=int(input("enter the lower limit : "))
upper = int(input("enter the upper limit :"))
for i in range(lower,upper):
  if (i%3==0):
    continue
  else:
    print(i*i)
  i= i+1
print(i)
```



```python
def shortestDistance(S, X):

    # Find distance from occurrences of X
    # appearing before current character.
    inf = float('inf')
    prev = inf
    ans = []
    for i,j in enumerate(S):
        if S[i] == X:
            prev = i
        if (prev == inf) :
            ans.append(inf)
        else :
            ans.append(i - prev)


    # Find distance from occurrences of X
    # appearing after current character and
    # compare this distance with earlier.
    prev = inf
    for i in range(len(S) - 1, -1, -1):
        if S[i] == X:
            prev = i
        if (X != inf):
            ans[i] = min(ans[i], prev - i)

    # return array of distance
    return ans
```

```
[0, 1, 2, 3, 4, 3, 2, 1, 0, 1, 2, 3, 4]
>
```

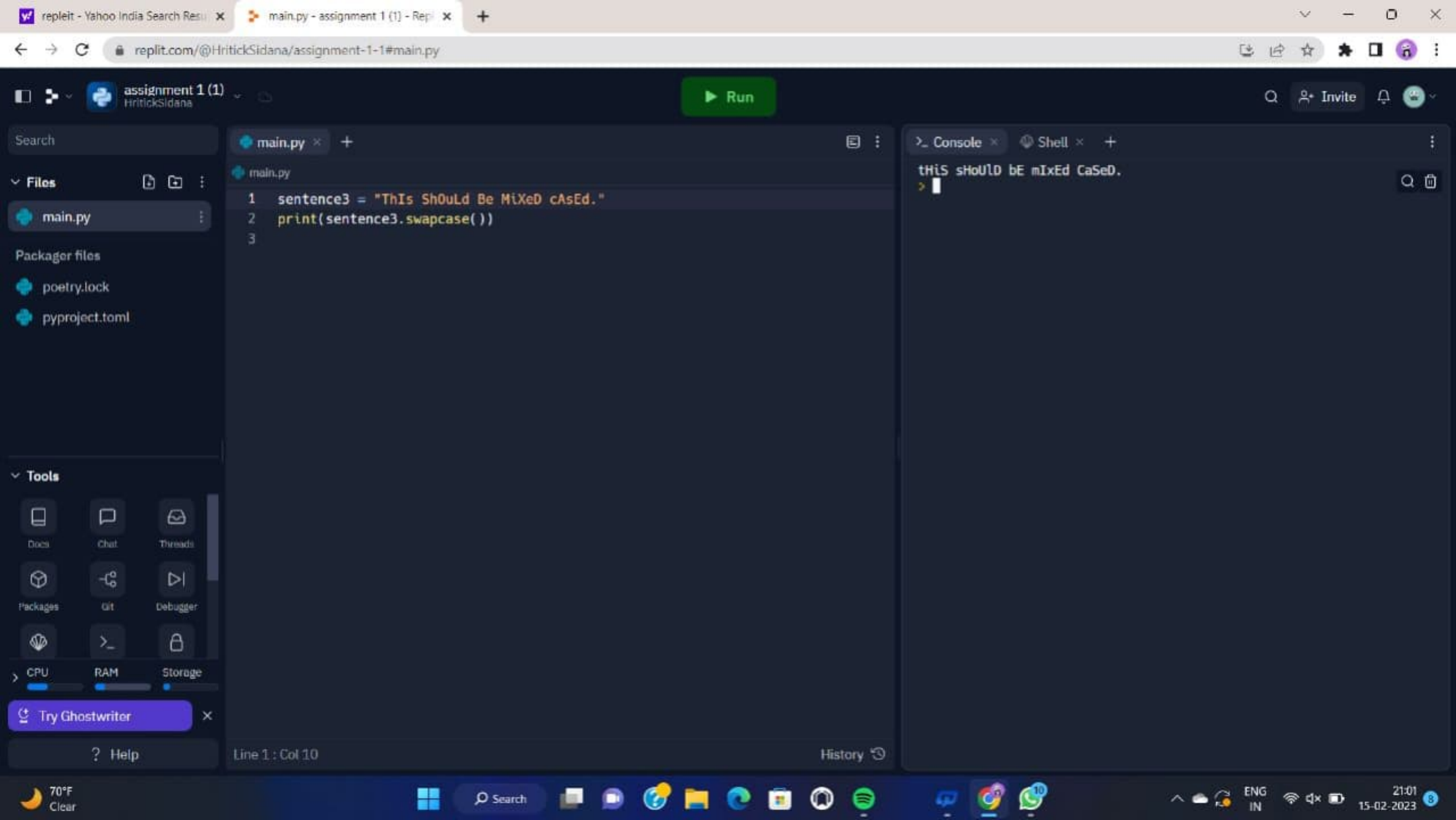
```
string1=input("enter the first string :")
string2=input("enter the second string :")
string1= string1.replace('A','a')
string2= string2.replace('A','a')
print(string1 + string2)
```



```python
def Check_Vow(string, vowels):
    string = string.casefold()
    count = {}.fromkeys(vowels, 0)

    for character in string:
        if character in count:
            count[character] += 1
    return count
vowels = 'aeiou'
string = input("enter the string : ")
print (Check_Vow(string, vowels))
```

```
enter the string : hritick
{'a': 0, 'e': 0, 'i': 2, 'o': 0, 'u': 0}
>
```

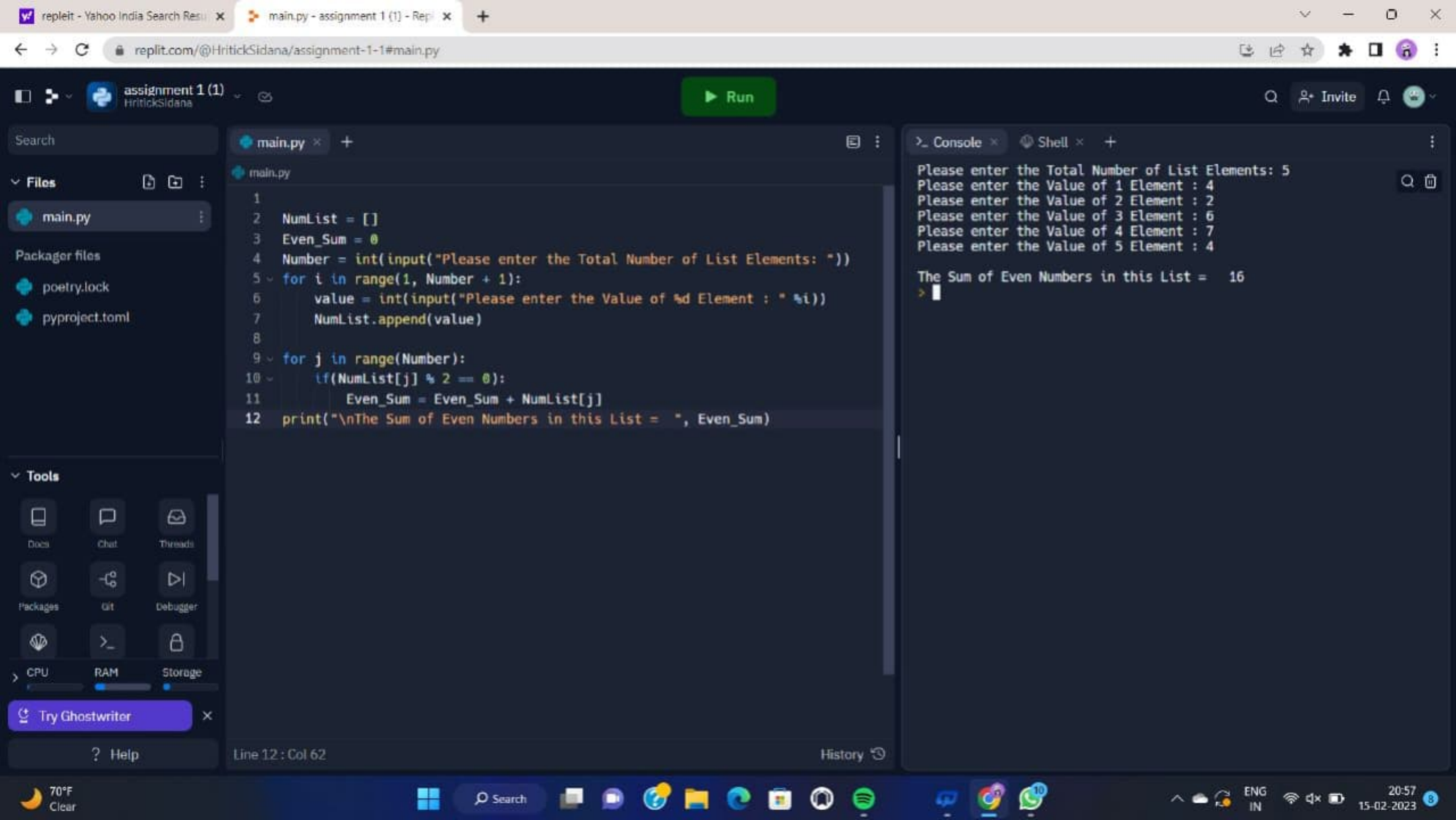
```
list1 = [10, 21, 4, 45, 66, 93]

only_odd = [num for num in list1 if num % 2 == 1]

print(only_odd)
```



```python
sentence3 = "ThIs ShOuLd Be MiXeD cAsEd."
print(sentence3.swapcase())
```

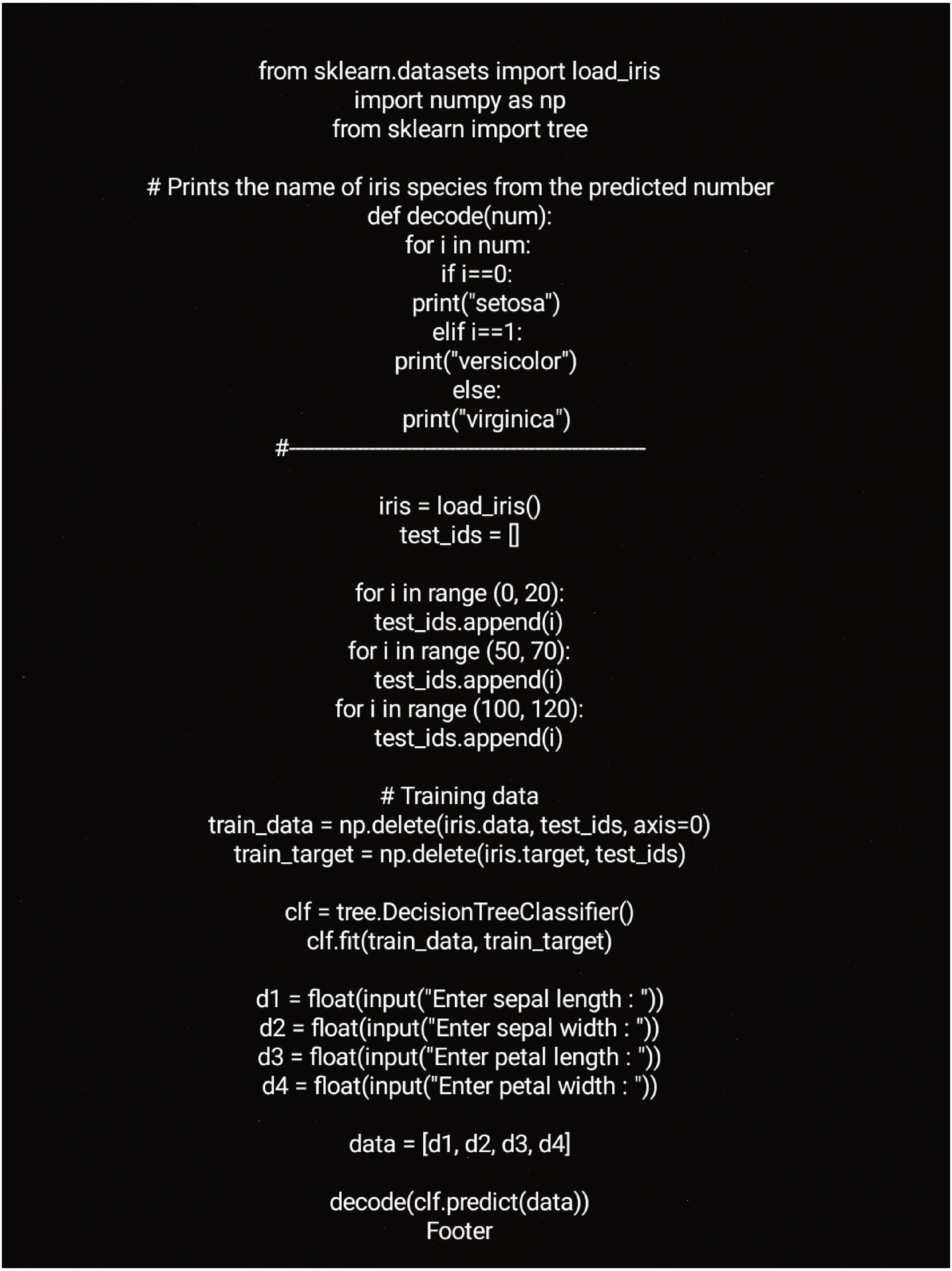
```
tHiS sHoUlD bE mIxEd CaSeD.
>
```

```
sentence3 = "ThIs ShOuLd Be MiXeD
cAsEd."
print(sentence3.swapcase())
```



```python

NumList = []
Even_Sum = 0
Number = int(input("Please enter the Total Number of List Elements: "))
for i in range(1, Number + 1):
    value = int(input("Please enter the Value of %d Element : " %i))
    NumList.append(value)

for j in range(Number):
    if(NumList[j] % 2 == 0):
        Even_Sum = Even_Sum + NumList[j]
print("\nThe Sum of Even Numbers in this List =  ", Even_Sum)
```

```
Please enter the Total Number of List Elements: 5
Please enter the Value of 1 Element : 4
Please enter the Value of 2 Element : 2
Please enter the Value of 3 Element : 6
Please enter the Value of 4 Element : 7
Please enter the Value of 5 Element : 4

The Sum of Even Numbers in this List =   16
>
```



```
from sklearn.datasets import load_iris
import numpy as np
from sklearn import tree

# Prints the name of iris species from the predicted number
def decode(num):
for i in num:
if i==0:
print("setosa")
elif i==1:
print("versicolor")
else:
print("virginica")
#---------------------------------------------------

iris = load_iris()
test_ids = []

for i in range (0, 20):
test_ids.append(i)
for i in range (50, 70):
test_ids.append(i)
for i in range (100, 120):
test_ids.append(i)

# Training data
train_data = np.delete(iris.data, test_ids, axis=0)
train_target = np.delete(iris.target, test_ids)

clf = tree.DecisionTreeClassifier()
clf.fit(train_data, train_target)

d1 = float(input("Enter sepal length : "))
d2 = float(input("Enter sepal width : "))
d3 = float(input("Enter petal length : "))
d4 = float(input("Enter petal width : "))

data = [d1, d2, d3, d4]

decode(clf.predict(data))
Footer
```