WEEK-12-MCQ-Files

1.Which of the following methods is used to read a single line from a file?

a.readline() b.readline(size) c.read() d.readlines()

2.Which of the following is data file?

a.none of the given choices b.Both text and binary file

c.Binary file d.Text file

3.How do you check if a file is closed in Python?

a.file.closed b.file.closed() c.file.isclosed() d.file.is_closed()

4.To open a file a.txt for reading, we use _____

a.infile = open(file = “a.txt”, “r”)

b. infile = open(“a.txt”, “r”)

c.infile = open(“a.txt”, “r”)

d.infile = open(file = “a.txt”, “r”)

5.what happens if you try to open a file in read mode ('r') that does not exist?

a.An error is raised b.The operation is ignored

c.An empty file object is returned d.A new file is created

6.Which method is used to close a file in Python?

a.terminate() b.end() c.close() d.finish()

7.What does the flush() method do in file handling?

a.Moves the file cursor to the beginning

b.Deletes the file content

c.Closes the file

d.Flushes the internal buffer to the file

8.Which of the following modes opens a file for writing in binary format?

a.rb b.r+ c.wb d.w

9.What does the following code do?

fo = open("foo.txt", "w")

a.Opens an existing file in read mode b.Opens a file in binary mode

c.Creates a new file for writing d.Appends to an existing file

10.What will the read() method return if used on an empty file?

a.None b.An empty string c.EOFError d.0

11.What is the difference between r+ and w+ modes in file handling?

a.r+ is for binary read and write, w+ is for text read and write

b.r+ is for read and write, w+ is for append only

c.r+ is for read and append, w+ is for read and write

d.r+ is for read and write without truncating, w+ is for read and write with truncating

12.What will be the output of the following code if the file "example.txt" contains "Python Programming"?

```
f = open("example.txt", "r")
print(f.read(6)) f.close()
```

a.An error is raised b.Python c.Python Programming d.Program

13. file_object.seek(offset [, reference_point])

What is the default, the value of reference_point?

a.1 b.0 c.2 d.null e.garbage

14.What happens if you try to open a file in write mode ('w') and the file already exists?

a.It does nothing b.It raises an error

c.It appends data to the file d.It clears the old data and starts fresh

15.How can you read a file line by line efficiently in Python?

a.Using the readline() method b.Using the readlines() method

c.Using a for loop d.Using the read() method

Write a Python program to count the frequency of each word in a given text file.

Description:

1. Input:
    - String as input.
2. Output:
    - A list of words with their corresponding frequency count to be write in a file "output.txt"

Example:

- Input File Content:

apple orange apple banana apple orange

Output:

apple: 3
orange: 2
banana: 1

For example:

Test	Input	Result

`with open('output.txt', 'r') as file:`
`  text = file.read()`
`  print(text)`
	`apple orange apple banana apple orange`	`apple: 3`
		`banana: 1`
		`orange: 2`

Answer:(penalty regime: 0 %)

Feedback

```
n=input()
l=''.join([c for c in n if c.isalnum() or c.isspace()])
l=l.split()
l=sorted(l,key=str.lower)
d={}
for i in l:
    d[i.lower()]=str(l.count(i))
s=''
for i in d:
    s+=(i+': '+d[i]+'\n')
f=open('output.txt','w')
f.write(s)
f.close()
```

	Test	Input

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		`apple orange apple banana apple orange`

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		`Hello world! Hello everyone. Welcome to the world of prog`

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		`One fish two fish Red fish blue fish`

Create a Python program to delete a specific line from a text file based on a given line number.

Description:

1. Input:
    - A text file with multiple lines.
    - A line number to delete.
2. Output:
    - The updated file with the specified line removed in file "output.txt".

Example:

- Input File Content:
  "Line one.
  Line two.
  Line three.
  Line four."
  2

  Updated line two.

Output:

Line one.
Line three.
Line four.

For example:

Test	Input	Result

`with open('output.txt', 'r') as file:`
`  text = file.read()`
`  print(text)`
	input1.txt
	2
		Line one.
		Line three.
		Line four.

```
fi=input()
n=int(input())
o='output.txt'
with open(fi,'r') as f:
  l=f.readlines()
l.remove(l[n-1])
with open(o,'w')as f:
  f.writelines(l)
```

	Test	Input	Expected	Got	

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		input1.txt
		2
			Line one.
			Line three.
			Line four.
				Line one.
				Line three.
				Line four.
					✔

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		input2.txt
		3
			Line A.
			Line B.
				Line A.
				Line B.
					✔

Passed all tests! ✔

Correct

Develop a Python program to copy the contents of one file to another file.

Description:

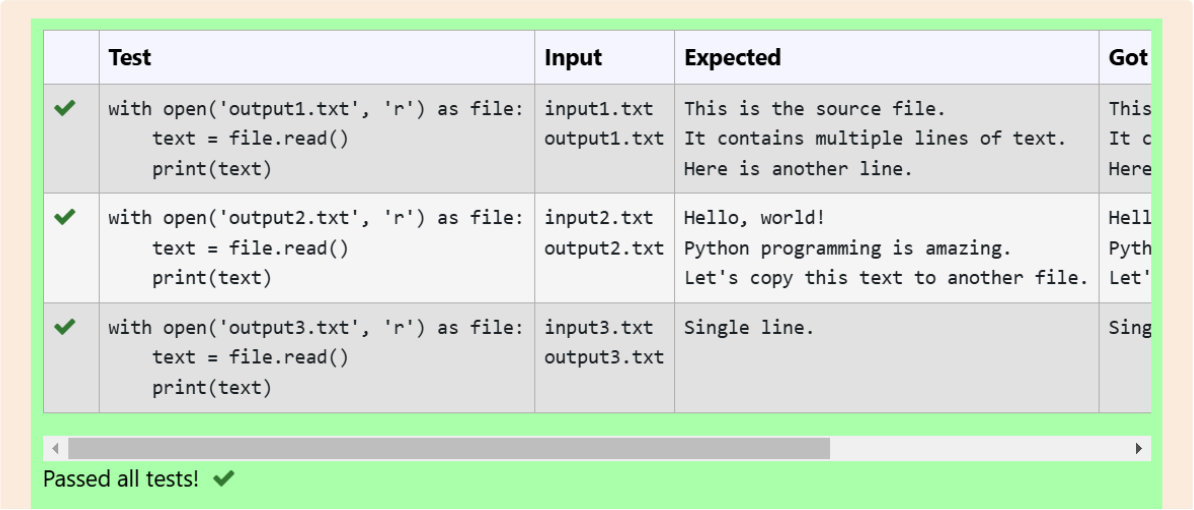
1. Input:
   - Source file and destination file names.
2. Output:
   - The content of the source file copied to the destination file.

For example:

Test	Input	Result

`with open('output1.txt', 'r') as file:`
`  text = file.read()`
`  print(text)`
	`input1.txt`
	`output1.txt`
		`This is the source file.`
		`It contains multiple lines of text.`
		`Here is another line.`

```
i=input()
o=input()
with open(i,'r')as f:
   with open(o,'a')as f1:
      f1.write(f.read())
```



	Test	Input	Expected	Got

✔	`with open('output1.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		`input1.txt`
		`output1.txt`
			`This is the source file.`
			`It contains multiple lines of text.`
			`Here is another line.`
				`This`
				`It c`
				`Here`

✔	`with open('output2.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		`input2.txt`
		`output2.txt`
			`Hello, world!`
			`Python programming is amazing.`
			`Let's copy this text to another file.`
				`Hell`
				`Pyth`
				`Let'`

✔	`with open('output3.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		`input3.txt`
		`output3.txt`
			`Single line.`	`Sing`

Passed all tests! ✔

Write a Python program to reverse the contents of a specific line in a text file based on a given line number.

Description:

1. Input:
   - A text file with multiple lines.
   - A line number to reverse.
2. Output:
   - The updated file with the specified line's contents reversed in file "output.txt".

Example:

- Input File Content:
  "Line one.
  Line two.
  Line three.

Line four."
3

Output:

Line one.
Line two.
eerht eniL.
Line four.

For example:

Test	Input	Result

`with open('output.txt', 'r') as file:`
`  text = file.read()`
`  print(text)`
	input1.txt
	3
		Line one.
		Line two.
		eerht
		eniL.
		Line four.

```
f1=input()
n=int(input())
with open(f1,'r')as f:
    l=f.readlines()
l[n-1]=l[n-1][::-1]
l[n-1]=l[n-1][2:]+'.\n'
with open('output.txt','w') as f:
    f.writelines(l)
```

	Test	Input	Expected	Got	

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		input1.txt
		3
			Line one.
			Line two.
			eerht eniL.
			Line four.
				Line one.
				Line two.
				eerht eniL.
				Line four.
					✔

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		input2.txt
		2
			Line A.
			B eniL.
			Line C.
				Line A.
				B eniL.
				Line C.
					✔

Passed all tests! ✔

Create a Python program to write to a specific line in a text file, replacing the existing content of that line.

Description:

1. Input:
    - A text file with multiple lines.
    - A line number to write to.
    - New content for the specified line.
2. Output:
    - The updated file with the specified line replaced by the new content in file "output.txt".

Example:

- Input File Content:
  "Line one.
  Line two.
  Line three.
  Line four."
  2

  Updated line two.

Output:

Line one.
Updated line two.
Line three.
Line four.

For example:

Test	Input	Result

`with open('output.txt', 'r') as file:`
`    text = file.read()`
`    print(text)`
	`input1.txt`
	`2`
	`Updated line two.`
		`Line one.`
		`Updated line two.`
		`Line three.`
		`Line four.`

```
i=input()
n=int(input())
```

```
s=input()
with open(i,'r')as f:
    l=f.readlines()
l[n-1]=s+'\n'
with open('output.txt','w')as f:
 f.writelines(l)
```

	Test	Input	Expected	Got

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		input1.txt
		2
		Updated line two.
			Line one.
			Updated line two.
			Line three.
			Line four.
				Line one.
				Updated line two.
				Line three.
				Line four.

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		input2.txt
		2
		Line B Updated.
			Line A.
			Line B Updated.
			Line C.
				Line A.
				Line B Updated.
				Line C.

Passed all tests! ✔

Develop a Python program to identify and print all palindrome words from a given text file.

Description:

1. Input:
    - A text file containing multiple words.
2. Output:
    - A list of palindrome words found in the file name as 'output.txt'.

For example:

Test	Input	Result

`with open('output.txt', 'r') as file:`
`  text = file.read()`
`  print(text)`
	input1.txt	madam
		arora
		malayalam

```
i=input()
with open(i,'r') as f:
```

```
    l=f.read()
l=l.split()
s=''
for i in l:
    if i==i[::-1]:
        s+=i+'\n'
with open('output.txt','w')as f:
    f.write(s)
```

	Test	Input	Expected	Got	

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		input1.txt	madam
			arora
			malayalam
				madam
				arora
				malayalam
					✔

Passed all tests! ✔

Develop a Python program to read a text file and count the total number of words in the file.

Description:

1. Input:
    - A text file containing several lines of text.
    - File name you should get as input.
2. Output:
    - The total number of words in the file.

For example:

Input	Result
input2.txt	Total words: 14
input3.txt	Total words: 15

```
import re
i=input()
```

```
with open(i,'r')as f:
    l=f.readlines()
l1=[]
for i in l:
    x=i.split()
    l1.extend(x)
print('Total words:',len(l1))
```

	Input	Expected	Got	
✔	input1.txt	Total words: 6	Total words: 6	✔
✔	input2.txt	Total words: 14	Total words: 14	✔
✔	input3.txt	Total words: 15	Total words: 15	✔

Passed all tests! ✔

Develop a Python program to read a specific line from a text file based on a given line number.

Description:

1. Input:
    - A text file with multiple lines.
    - A line number to read.
2. Output:
    - The content of the specified line.

input1.txt:

Line one.
Line two.
Line three.
Line four.

For example:

Input	Result

input1.txt 3	Line three.

```
i=input()
n=int(input())
with open(i,'r')as f:
    l=f.readlines()
print(l[n-1])
```

	Input	Expected	Got	
✔	input1.txt 3	Line three.	Line three.	✔
✔	input2.txt 3	Line C.	Line C.	✔

Passed all tests! ✔

Write a Python program to append a new line at a specific position in a text file, shifting existing lines down.

Description:

1. Input:
    - A text file with multiple lines.
    - A line number to insert the new line at.
    - New content for the new line.
2. Output:
    - The updated file with the new line inserted at the specified position, shifting the existing lines down in file "output.txt".

Example:

- Input File Content:
  "Line one.
  Line two.
  Line three.

Line four."
3

Inserted line..

Output:

Line one.
Line two.
Inserted line.
Line three.
Line four.

For example:

Test	Input	Result

`with open('output.txt', 'r') as file:`
`  text = file.read()`
`  print(text)`
	`input1.txt`
	`3`
	`Inserted line.`
		`Line one.`
		`Line two.`
		`Inserted line.`
		`Line three.`
		`Line four.`

```
i=input()
n=int(input())
s=input()
s+='\n'
with open(i,'r')as f:
    l=f.readlines()
if n-1==len(l):
    l[-1]+='\n'
l.insert(n-1,s)
with open('output.txt','w')as f:
    f.writelines(l)
```

	Test	Input	Expected	Got	

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		input1.txt
		3
		Inserted line.
			Line one.
			Line two.
			Inserted line.
			Line three.
			Line four.
				Line one.
				Line two.
				Inserted line.
				Line three.
				Line four.
					✔

✔	`with open('output.txt', 'r') as file:`
	`  text = file.read()`
	`  print(text)`
		input2.txt
		4
		Inserted line D.
			Line A.
			Line B.
			Line C.
			Inserted line D.
				Line A.
				Line B.
				Line C.
				Inserted line D.
					✔

Passed all tests! ✔

Create a Python program to find the longest word in a text file.

- Input:
  - A text file containing multiple lines of text.
- Output:
  - The longest word in the file.

For example:

Input	Result
input1.txt	Longest word: containing

```
i=input()
with open(i,'r') as f:
    l=f.read()
l=l.split()
s=max(l,key=len)
print('Longest word:',s)
```

	Input	Expected	Got
✔	input1.txt	Longest word: containing	Longest word: containing
✔	input2.txt	Longest word: thousand	Longest word: thousand
✔	input3.txt	Longest word: supercalifragilisticexpialidocious	Longest word: supercalifragilistic

Passed all tests! ✔