**Python and OOP ( Theory-Mid ) Answer Script**

| Question No. 01` |
| --- |
|  |
| Answer:  def check\_palindrome(text):      rev\_string=text[::-1]      # print(rev\_string)      if text==rev\_string:          print("palindrom")      else:          print("not palindrome")  # main()  text="dppafappd"  check\_palindrome(text)  text="python"  check\_palindrome(text) |

| Question No. 02 |
| --- |
|  |
| Answer:  # file read  with open('align.txt','r') as file:  lines=file.readlines()  # print(lines) # all lines are in a single list  # split and append to an empty list  list=[]  for i in lines:  # print(i)  list.append(i.split("\n")) # split when newline is found  # print(list) # every lines are in seperate list,we need the 0'th index  # taking the 0th element and center align  for i in list:  # print(i[0])  print('{:^80s}'.format(i[0])) # technique 1  # print(i[0].center(80)) # technique 2 |

| Question No. 03 |
| --- |
|  |
| Answer:  def nearly\_equal(text1, text2):  found\_chars=[]  for i in text1:  for j in text2:  if i==j:  # print(i,j,end=" ")  found\_chars.append(i)  # print(found\_chars)  # print(len(text1), len(found\_chars), len(text2))  if len(text1) == len(found\_chars):  print(True)  else:  print(False)  # main()  nearly\_equal('perl', 'pearl')  nearly\_equal('python', 'perl') |

| Question No. 04 |
| --- |
|  |
| Answer:  def anagrams(list\_s):  anagrams\_list={}  for s in list\_s:  # print(s)  sorted\_s=str(sorted(s)).lower()    # print(sorted\_s)  # print(sorted\_s, s)  if sorted\_s in anagrams\_list:  anagrams\_list[sorted\_s].append(s)  else:  anagrams\_list[sorted\_s]=[s]  return list(anagrams\_list.values())    # main()  list\_s=['eat', 'ate', 'done', 'tea', 'soup', 'node']  res=anagrams(list\_s)  print(res) |

| Question No. 05 |
| --- |
|  |
| Answer:  global names= space\_invaders, player\_pos, level, max\_level  built in names = def, while, if, len, continue |

| Question No. 06 |
| --- |
|  |
| Answer:  The length of l1 is 3, so for loop runs 3 times.   | i | return of p(*l1,l2,i*) | r=r+ p(*l1,l2,i*) | | --- | --- | --- | | 0 | 4\*1= 4 | r=0+4=4 | | 1 | 2\*5=10 | r=4+10=14 | | 2 | 3\*6= 18 | r=14+18=32 |   Final result r= 32 |

| Question No. 07 |
| --- |
|  |
| Answer:  import keyboard  # file read  with open('File.txt','r') as file:  lines=file.readlines()  # print(lines)  # print(lines)  # converting to string from list  s=''  for i in lines:  s+=i  # print(s)  # split when "--" is found  res=s.split("--")  # print(res)  # read key press  for w in res:  print(w)  print("[enter - read more, press q to quit]")  # if keyboard.read\_key() == "enter":  keyboard.read\_key() # user key press  if keyboard.read\_key() == "enter":  continue  else:  print("quit")  break |

| Question No. 08 |
| --- |
|  |
| Answer:  with open('File.txt','r') as file:  lines=file.readlines()  # converting to string from list  s=''  for i in lines:  s+=i  # print(s)  # split when "--" is found  res=s.split("--")  # print(res)  # main operation  i=0  while True:  print(res[i])  num=input("Enter a page number to open(forward), press enter for next page, q to quit: ")  # print(num, type(num)) # num- string class  if num=="q":  print("quit")  break    elif num.isdigit():  num=int(num)  # print(num, type(num)) # num- int  # print(res[num])  i=num+i  else:  # print(i, res[i+1]) # type string  i=i+1 |

| Question No. 09 |
| --- |
|  |
| Answer:  with open('File.txt','r') as file:  lines=file.readlines()  # converting to string from list  s=''  for i in lines:  s+=i  # print(s)  # split when "--" is found  res=s.split("--")  # print(res)  # main operation  i=0  while True:  # for w in res:  print(res[i])  num=input("Enter a page number to open(forward and backward), press enter for next page, q to quit: ")  # print(num, type(num)) # num- string class  if num=='q': # for quit  print("quit")  break  elif num=="": # for enter  i=i+1  # print("e", i)  elif num.isdigit(): # for pos  num=int(num)  # print(num, type(num)) # num- int  # print(res[num])  i=num+i  # print("p", i)  else: # for negative  # print(i, res[i+1]) # type string  num=int(num)  i=num+i  # print("n", i) |

| Question No. 10 |
| --- |
|  |
| Answer:  Files with the same names cannot reside in the same folder, files with the same name can be created in different folders. Because a file name is a unique identifier for a particular file. If two files with the same name are created in the same folder, OS cannot separate them as different files. so, OS can maintain separate files with the same if they are in different folders,that means files are in different locations. |

Submitted by: Subrata Saha

Email:[Subratabaec@gmail.com](mailto:Subratabaec@gmail.com)

Date:05.11.2022

THE END