Assignment_2_function

February 23, 2023

[ ]:
```
Q1. Which keyword is used to create a function? Create a function to return a␣
 ↪list of odd numbers in the
range of 1 to 25.
```

[ ]:
```
ans:-def keyword is used to create function.
```

[6]:
```
def odd_list():
    for i in range(1,25):
        if i % 2 !=0 :
                print(i)
```

[ ]:
```
Q2. Why *args and **kwargs is used in some functions? Create a function each␣
 ↪for *args and **kwargs
to demonstrate their use.
```

[ ]:
```
ans:=*args means n number of argument passed.
    **args used in dictory values.
```

[8]:
```
def test(*args):
    return args
```

[10]:
```
test(2,3,4)
```

[10]:
```
(2, 3, 4)
```

[11]:
```
def test(**args):
    return args
```

[12]:
```
type(test())
```

[12]:
```
dict
```

[ ]:
```
Q3.  What is an iterator in python? Name the method used to initialise the␣
 ↪iterator object and the method
used for iteration. Use these methods to print the first five elements of the␣
 ↪given list [2, 4, 6, 8, 10, 12, 14,
```

```
16, 18, 20].
```

[ ]:
```
Ans:-An iterator is an object that contains a countable number of values.
     method used to initilaise the iterator object __iter__() and __next__().
```

[29]:
```
l=[2, 4, 6, 8, 10, 12, 14,16, 18, 20]
list=iter(l)
print(next(list))
print(next(list))
print(next(list))
print(next(list))
print(next(list))
```

```
2
4
6
8
10
```

[ ]:
```
Q4.  What is a generator function in python? Why yield keyword is used? Give an␣
 ↪example of a generator
function.
ans:-generator is like a normal function it used to generate value
     yeild keyword used for return values
```

[30]:
```
def gener():
    yield 1
    yield 2
    yield 3
```

[31]:
```
for i in gener():
    print (i)
```

```
1
2
3
```

[ ]:
```
Q5. Create a generator function for prime numbers less than 1000. Use the␣
 ↪next() method to print the
first 20 prime numbers.
```

[ ]:
```
def prime():
    l1=[]
    for i in range(1,1001):
        for num in range(2,i):
            if i % num ==0:
                break
```

```
            ## num=i+1
        else :
            l1.append(i)
            l=iter(l1)
            print(next(l))
```

[ ]:
```
Q6. Write a python program to print the first 10 Fibonacci numbers using a␣
↪while loop.
```

[2]:
```
a,b=0,1
counter=0
while counter<10:
    print(a)
    c=a+b
    a=b
    b=c
    counter=counter+1
```

```
0
1
1
2
3
5
8
13
21
34
```

[ ]:
```
Q7. Write a List Comprehension to iterate through the given string: ‘pwskills’.
Expected output: ['p', 'w', 's', 'k', 'i', 'l', 'l', 's'].
```

[12]:
```
[ l for l in 'pwskills']
```

[12]:
```
['p', 'w', 's', 'k', 'i', 'l', 'l', 's']
```

[ ]:
```
Q8. Write a python program to check whether a given number is Palindrome or not␣
↪using a while loop
```

[26]:
```
num = int(input('please enter number'))
temp = num
reverse = 0
while temp > 0:
    remainder = temp % 10
    reverse = (reverse * 10) + remainder
    temp = temp // 10
```

```
if num == reverse:
    print('Palindrome')
else:
    print("Not Palindrome")
```

```
please enter number 121

Palindrome
```

[ ]:
```
Q9. Write a code to print odd numbers from 1 to 100 using list comprehension
```

[29]:
```
num_list=[i for i in range(1,101)]
```

[36]:
```
odd_list=[i for i in num_list  if i%2!=0 ]
```

[37]:
```
odd_list
```

[37]:
```
[1,
 3,
 5,
 7,
 9,
 11,
 13,
 15,
 17,
 19,
 21,
 23,
 25,
 27,
 29,
 31,
 33,
 35,
 37,
 39,
 41,
 43,
 45,
 47,
 49,
 51,
 53,
 55,
 57,
 59,
 61,
```

```
63,
65,
67,
69,
71,
73,
75,
77,
79,
81,
83,
85,
87,
89,
91,
93,
95,
97,
99]
```

[ ]: