Assignment 6

Assignment not submitted

Due date: 2020-03-11, 23:59 IST.

Give that the statement chr(ord(alpha) + i) returns the character(alphabet or a special character) at the location i ahead than the alphabet alpha, eg, chr(ord('a')+1) returns 'b'; what is the output of the following code? ***1 point***

```
def encrypt(ltr,key):
  l=[]
  for each in list(ltr):
    l.append(chr(ord(each) + 1))
  return ("".join(l))

letter_body="ABCDEFGH"
d=encrypt(letter_body,4)
print(d)
```

- ABCDEFGH
- BCDEFGHI
- EFGHIJKL
- none of the above

What does the following code do? ***1 point***

```
def guess(num):
  a=input("Guess a number")
  if(a==num):
    print("SUCCESS")
  else:
    guess(num)

guess(10)
```

- Keeps asking the user to guess a number until the user guesses 10
- The computer generates a random number r and keeps it. The user is repeatedly prompted to enter a number. If the user enters r, the code says success and ends, else the prompting is continued.
- Enters an infinite loop
- The computer generates a random number r and keeps it. The user is repeatedly prompted to enter a number. If the user enters r, the code says success and ends, else the computer generates a new random number r and thereafter the prompting is continued.

What does the following code do? ***1 point***

```
import random
def guess(num):
  a=int(input("Guess a number from 1 to 100"))
  print(a,num)
  if(a==num):
    print("SUCCESS")
  else:
    guess(random.randint(1,100))

i=guess(random.randint(1,100))
```

- Keeps asking the user to guess a number until the user guesses 10
- The computer generates a random number r and keeps it. The user is repeatedly prompted to enter a number. If the user enters r, the code says success and ends, else the prompting is continued.
- Enters an infinite loop
- The computer generates a random number r and keeps it. The user is repeatedly prompted to enter a number. If the user enters r, the code says success and ends, else the computer generates a new random number r and thereafter the prompting is continued.

With n as input, the code below computes ***1 point***

```
def mul(num):
  if(num==1):
    return(-1)
  return(-1*mul(num-1))

n=int(input("Enter the value of n"))
print(mul(n))
```

- $-1 \times n$
- $-1 + n$
- $(-1)^n$
- $n^{(-1)}$

The following code ***1 point***

```
import random
def search(l,loc,item):
  if(loc<0):
    loc=0
  if(l[loc]==item):
    print("Found",item,"at index",loc)
    return
  if(loc==len(l)-1):
    print("Element not present")
    return(0)
  else:
    return(search(l,loc+1,item))

l=[1,2,3,4,5,6,7,8,9]
search(l,-11,3)
```

- displays an error
- does not display an error but might display the error if we change the middle value passed in the function search() from 0 to some negative value.
- Can return a negative value in some cases when we change the values passed to the function search()
- Scans the list from first to the last element and displays the index of the value passed in the last number in the function search().

The following code represents **1 point**

```
import random
def search(l,loc,item):
    if(loc<0):
        loc=0
    if(l[loc]==item):
        print("Found",item,"at index",loc)
        return
    if(loc==len(l)-1):
        print("Element not present")
        return(0)
    else:
        return(search(l,loc+1,item))

l=[1,2,3,4,5,6,7,8,9]
search(l,-11,3)
```

- recursive algorithm for linear search an element in a list
- recursive algorithm for binary search an element in a list
- non-recursive algorithm for linear search an element in a list
- none of the above

What is the output of print(int(3.79)+int(2.1))? **1 point**

- 6
- 5
- 7
- 4

The following code to its best, represents a scenario **1 point**

```
def func(i):
    print(i)
    if(i==0):
        print("OVER")
    else:
        func(i/2)
```

- A cake getting eaten by half of its current amount every time
- A student attempting alternate questions, starting from a given question
- Viruses doubling inside a body and killing the person once their population becomes 128 or more.
- Metro train serving 128 stations to and fro

The following code to its best, represents a scenario **1 point**

```
def func(i):
  print(i)
  if(i>128):
    print("OVER")
  else:
    func(2*i)
```

- A cake getting eaten by half of its current amount every time
- A student attempting alternate questions, starting from a given question
- Viruses doubling inside a body and killing the person once their population becomes 128 or more.
- Metro train serving 128 stations to and fro

The following code to its best, represents a scenario ***1 point***

```
def func(i,f):
  print(i)
  if(i==0):
    f=1
    func(i+1,f)
  if(i==128):
    f=-1
    func(i-1,f)
  if(f==1):
    func(i+1,f)
  if(f==-1):
    func(i-1,f)
```

- A cake getting eaten by half of its current amount every time
- A student attempting alternate questions, starting from a given question
- Viruses doubling inside a body and killing the person once their population becomes 128 or more.
- Metro train serving 128 stations to and fro

Submit Answers

number of times before the due date. The final submission will be considered for grading.