

X


<https://swayam.gov.in>

https://swayam.gov.in/nc_details/NPTEL

reviewer4@nptel.iitm.ac.in ▾

[NPTEL \(https://swayam.gov.in/explorer?ncCode=NPTEL\)](https://swayam.gov.in/explorer?ncCode=NPTEL) » [The Joy of Computing using Python \(course\)](#)
[Announcements \(announcements\)](#) [About the Course \(https://swayam.gov.in/nd1_noc20_cs35/preview\)](https://swayam.gov.in/nd1_noc20_cs35/preview)
[Ask a Question \(forum\)](#) [Progress \(student/home\)](#) [Mentor \(student/mentor\)](#)

Unit 9 - Week 7

Course outline

How does an NPTEL online course work?

Week 0

Week 1

Week 2

Week 3

week 4

Week 5

Week 6

Week 7

Snakes and Ladders - Not on the Board (unit? unit=121&lesson=122)

Snakes and Ladders - Not on the Board - Part 01 (unit? unit=121&lesson=123)

Assignment 7

The due date for submitting this assignment has passed. **Due on 2020-03-18, 23:59 IST.**
As per our records you have not submitted this assignment.

1) Predict the output

1 point

```

1 l=[[1,2,3],[4,5,6],[7,8,9]]
2 f=1
3 for j in range(3):
4     if (f==1):
5         for i in range(3):
6             print(l[i][j],end=" ")
7             f=0
8     if (f==0):
9         for i in range(2,-1,-1):
10            print(l[i][j],end=" ")
11            f=1

```

7 4 1 1 4 7 2 5 8 8 5 2 3 6 9 9 6 3

1 4 7 2 5 8 3 6 9

1 4 7 8 5 2 3 6 9

1 4 7 7 4 1 2 5 8 8 5 2 3 6 9 9 6 3

No, the answer is incorrect.
Score: 0

Accepted Answers:
1 4 7 7 4 1 2 5 8 8 5 2 3 6 9 9 6 3

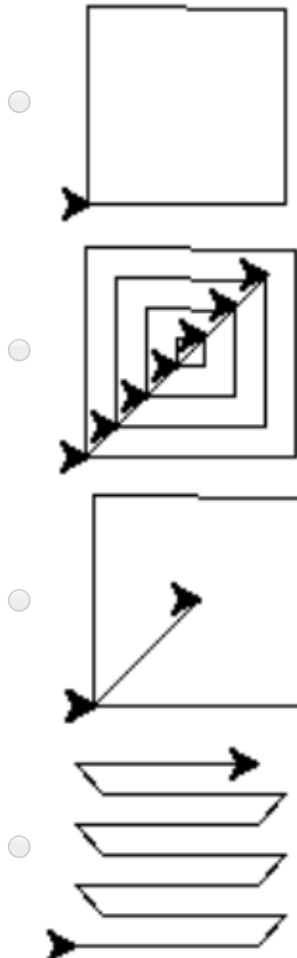
2) Predict the output of the calling function func1() for a given square matrix mx of dimension 70 × 70. **1 point**

- Snakes and Ladders - Not on the Board - Part 02 (unit? unit=121&lesson=124)
- Snakes and Ladders - Not on the Board - Part 03 (unit? unit=121&lesson=125)
- Snakes and Ladders - Not on the Board - Part 04 (unit? unit=121&lesson=126)
- Snakes and Ladders - Not on the Board - Part 05 (unit? unit=121&lesson=127)
- Snakes and Ladders - Not on the Board - Part 06 (unit? unit=121&lesson=128)
- Spiral Traversing - Let's Animate (unit? unit=121&lesson=129)
- Spiral Traversing - Let's Animate - Part 01 (unit? unit=121&lesson=130)
- Spiral Traversing - Let's Animate - Part 02 (unit? unit=121&lesson=131)
- Spiral Traversing - Let's Animate - Part 03 (unit? unit=121&lesson=132)
- Spiral Traversing - Let's Animate - Part 04 (unit? unit=121&lesson=133)
- Spiral Traversing - Let's Animate - Part 05 (unit? unit=121&lesson=134)

```

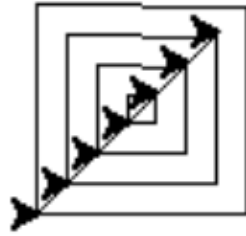
1 def func(mx, i):
2     tur = turtle.Turtle()
3     tur.setpos(i, i)
4
5     for ind in range(i, n-i):
6         tur.goto(i, ind)
7     for ind in range(i+1, n-i):
8         tur.goto(i, n-1-i)
9     for ind in range(n-2-i, i, -1):
10        tur.goto(n-1-i, ind)
11    for ind in range(n-i-1, i, -1):
12        tur.goto(ind, i)
13
14    def func1(mx):
15        n=len(mx)
16        i=0
17        while (i <=n-1):
18            func(mx, i)
19            i=i+10
20

```



No, the answer is incorrect.
 Score: 0
 Accepted Answers:

- Spiral Traversing - Let's Animate - Part 06 (unit? unit=121&lesson=135)
- Spiral Traversing - Let's Animate - Part 07 (unit? unit=121&lesson=136)
- GPS - Track the route (unit? unit=121&lesson=137)
- GPS - Track the route - Part 01 (unit? unit=121&lesson=138)
- GPS - Track the route - Part 02 (unit? unit=121&lesson=139)
- GPS - Track the route - Part 03 (unit? unit=121&lesson=140)
- GPS - Track the route - Part 04 (unit? unit=121&lesson=141)
- Quiz : Assignment 7 (assessment? name=277)**
- Programming Assignment-1: Lower Triangular Matrix (/noc20_cs35/progassignment? name=299)
- Programming Assignment-2: Symmetric (/noc20_cs35/progassignment? name=300)
- Programming Assignment-3: Binary Matrix (/noc20_cs35/progassignment? name=301)
- Week 7 Feedback (unit? unit=121&lesson=302)



3) Predict the output of the calling function func() for a given square matrix mx of dimension 70 × 70. **0 points**

```

1 def func(mx):
2     func1(mx,0)
3     tur = turtle.Turtle()
4     tur.setpos(0,0)
5     if ((len(mx))%2==1):
6         turtle.goto(int(len(mx)/2),int(len(mx)/2))
7     else:
8         second=int(len(mx)/2)
9         turtle.goto(second-1,second-1)
10        turtle.goto(second-1,second)
11        turtle.goto(second,second-1)
12        turtle.goto(second,second)
13

```

-
-
-
-

No, the answer is incorrect.
Score: 0

Week 8

Week 9

Week 10

Week 11

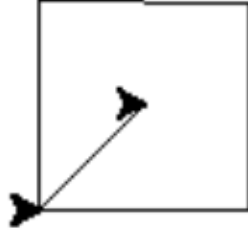
Week 12

Text Transcripts

Download Videos

Books

Accepted Answers:



4) Which of the following libraries is required to work with Google maps in Python?

1 point

- gplot
- googleplot
- gmpplot
- none of these

No, the answer is incorrect.

Score: 0

Accepted Answers:

gmpplot

5) Which of the following codes represent a correct version of a board game where the user has **1 point** to move from block 1 to block 100?

The game initialises only when the user gets a 1 or 6 on the dice and ends once he reaches 100 or gets a number which makes

him reach beyond 100 (i.e. the player wins if he is at 99 and gets a 4).

```

1 import random
2 def play(psn):
3     r = random.randint(1,6)
4     if(psn==0):
5         if(r==1 or r==6):
6             psn=1
7         else:
8             psn=psn+r
9     print("Position=",psn)
10    if(psn>=100):
11        print("You won")
12        return
13    play(psn)
14    position=0
15    print("Position=",position)
16    play(position)

```

```

1 import random
2
3 def play(psn):
4
5     r = random.randint(1,6)
6     print("Dice rolled:",r)
7     if (psn==0):
8         if (r==1 or r==6):
9             psn=1
10        else:
11            psn=psn+r
12        print("Position=",psn)
13        if (psn>=100):
14            print("You won")
15            return
16        play(psn)
17
18
19 position=0
20 print("Position=",position)
21 play(position)

```

```

1 import random
2
3 def play(psn):
4
5     r = random.randint(1,6)
6     print("Dice rolled:",r)
7     input()
8     if (psn==0):
9         if (r==1 or r==6):
10            psn=1
11        else:
12            psn=psn+r
13        print("Position=",psn)
14        if (psn>=100):
15            print("You won")
16
17        play(psn)
18
19
20 position=0
21 print("Position=",position)
22 play(position)

```

```

1 import random
2
3 def play(psn):
4

```

```

5  print("Dice rolled:",2)
6  if (psn==0):
7      psn=1
8  else:
9      psn=psn+2
10 print("Position=",psn)
11 if (psn>=100):
12     print("You won")
13
14 play(psn)
15 position=0
16 print("Position=",position)
17 play(position)

```

No, the answer is incorrect.

Score: 0

Accepted Answers:

```

1  import random
2
3  def play(psn):
4
5      r = random.randint(1,6)
6      print("Dice rolled:",r)
7      if (psn==0):
8          if (r==1 or r==6):
9              psn=1
10         else:
11             psn=psn+r
12         print("Position=",psn)
13         if (psn>=100):
14             print("You won")
15             return
16         play(psn)
17
18
19 position=0
20 print("Position=",position)
21 play(position)

```

6) Imagine a single player snakes and ladders game. The code below represents

1 point

```

1 import random
2
3 def play(psn):
4     snake_begin=-1
5     snake_end=-1
6     while(snake_begin <= snake_end):
7         snake_begin=random.randint(1,99)
8         snake_end=random.randint(1,99)
9         print("Snake from",snake_begin,"to",snake_end)
10        r = random.randint(1,6)
11        print("Dice rolled:",r)
12        if(psn==0):
13            if(r==1 or r==6):
14                psn=1
15            else:
16                psn=psn+r
17        print("Position=",psn)
18        input()
19        if(psn==snake_begin):
20            print("Bitten by snake")
21            psn=snake_end
22        if(psn>=100):
23            print("You won")
24            return
25        play(psn)
26
27 position=0
28 print("Position=",position)
29 play(position)

```

- A snakes and ladders game with one snake whose position remains constant while the player is playing. The position also remains the same during any subsequent plays (i.e. the game board does not change while you sleep and play again the next day).
- A snakes and ladders game with one snake whose position remains constant while the player is playing. However, the position can change during any subsequent plays (i.e. the game board might change while you sleep and play again the next day).
- A snakes and ladders game with one snake where the snake can change its position during the game and also during any subsequent plays (a board game where the snakes keep moving). Further, the snake can bite you any number of times.
- A snakes and ladders game with one snake where the snake can change its position during the game and also during any subsequent plays (a board game where the snake keeps moving). Further, the snake can bite you only ones when you play.

No, the answer is incorrect.

Score: 0

Accepted Answers:

A snakes and ladders game with one snake where the snake can change its position during the game and also during any subsequent plays (a board game where the snakes keep moving). Further, the snake can bite you any number of times.

7) Imagine a single player snakes and ladders game. The code below represents

1 point

```

1 import random
2
3 def play (psn , flag ):
4     snake_begin=-1
5     snake_end=-1
6     while (snake_begin <= snake_end):
7         snake_begin=random.randint (1 ,99)
8         snake_end=random.randint (1 ,99)
9         print ( " Snake from " , snake_begin , " to " , snake_end )
10        r = random.randint (1 ,6)
11        print ( " Dice rolled : " , r )
12        if (psn==0):
13            if (r==1 or r==6):
14                psn=1
15        else :
16            psn=psn+r
17        print ( " Position=" , psn )
18        #input ()
19        if (psn==snake_begin and flag ==0):
20            print ( " Bitten by snake " )
21            psn=snake_end
22            flag=1
23        if (psn >=100):
24            print ( " You won " )
25            return
26        play (psn , flag )
27
28 position=0
29 print ( " Position=" , position )
30 play ( position ,0)

```

- A snakes and ladders game with one snake whose position remains constant while the player is playing. The position also remains the same during any subsequent plays (i.e. the game board does not change while you sleep and play again the next day).
- A snakes and ladders game with one snake whose position remains constant while the player is playing. However, the position can change during any subsequent plays (i.e. the game board might change while you sleep and play again the next day).
- A snakes and ladders game with one snake where the snake can change its position during the game and also during any subsequent plays (a board game where the snakes keep moving). Further, the snake can bite you any number of times.
- A snakes and ladders game with one snake where the snake can change its position during the game and also during any subsequent plays (a board game where the snake keeps moving). Further, the snake can bite you only ones when you play.

No, the answer is incorrect.

Score: 0

Accepted Answers:

A snakes and ladders game with one snake where the snake can change its position during the game and also during any subsequent plays (a board game where the snake keeps moving). Further, the snake can bite you only ones when you play.

8) Assuming that the play1() function implements the recursive play of snakes and ladders with **1 point** the prespecified position of the snake and the ladder as shown in the code below, which of the ambiguities in the options can result in the code?

Kindly assume a typical snakes and ladders game.

```

1 import random
2 def play(psn):
3     snake_begin=-1
4     snake_end=-1
5     while(snake_begin <= snake_end):
6         snake_begin=random.randint(1,99)
7         snake_end=random.randint(1,99)
8     ladder_begin=-1
9     ladder_end=-1
10    while(ladder_end <= ladder_begin):
11        ladder_begin=random.randint(1,99)
12        ladder_end=random.randint(1,99)
13    play1(psn, snake_begin, snake_end, ladder_begin, ladder_end)

```

- snake_begin=snake_end
 ladder_begin=ladder_end
 ladder_begin=snake_begin
 ladder_end=snake_end

No, the answer is incorrect.

Score: 0

Accepted Answers:

ladder_begin=snake_begin

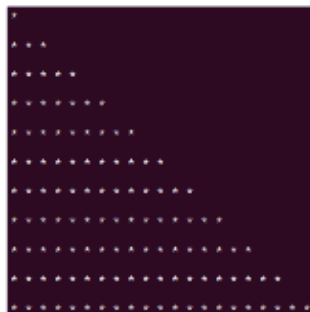
9) What is the output of the following code?

0 points

```

1 import random
2 def play(psn):
3     snake_begin=-1
4     snake_end=-1
5     while(snake_begin <= snake_end):
6         snake_begin=random.randint(1,99)
7         snake_end=random.randint(1,99)
8     ladder_begin=-1
9     ladder_end=-1
10    while(ladder_end <= ladder_begin):
11        ladder_begin=random.randint(1,99)
12        ladder_end=random.randint(1,99)
13    play1(psn, snake_begin, snake_end, ladder_begin, ladder_end)

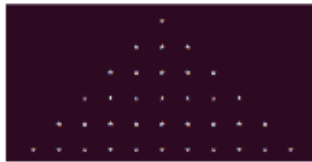
```



- 
- 
- 

No, the answer is incorrect.
Score: 0

Accepted Answers:



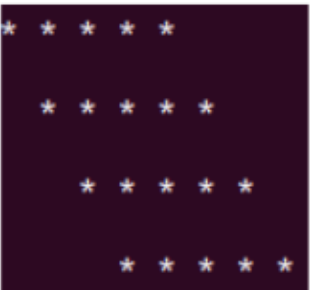
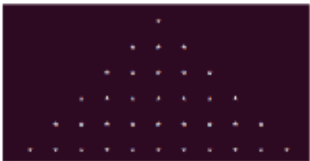
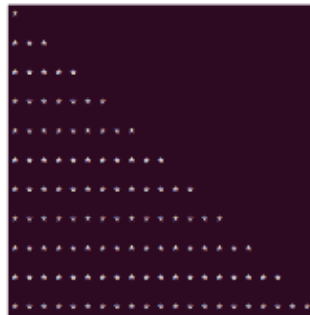
10)What is the output of the following code?

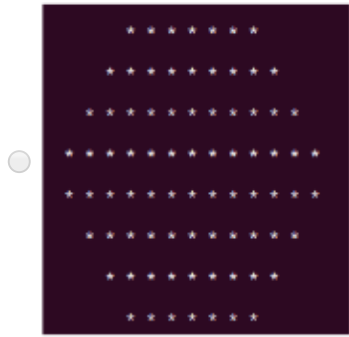
1 point

```

1
2 def func():
3     print()
4     c=10
5     i=3
6     while(i <=6):
7         j=0
8         while(j <=20):
9             if(j >=10-i and j <=10+i):
10                print('*',end=" ")
11            else:
12                print(' ',end=" ")
13            j=j+1
14            print('\n')
15            i=i+1
16        i=6
17        while(i >=3):
18            j=0
19            while(j <=20):
20                if(j >=10-i and j <=10+i):
21                    print('*',end=" ")
22                else:
23                    print(' ',end=" ")
24                j=j+1
25                print('\n')
26            i=i-1
27    func()

```





No, the answer is incorrect.
Score: 0

Accepted Answers:

