

Question Time

```
x = 1
if x>0:
    x = x-2
elif x<0:
    x = x+3
else:
    x = 0
```

What is the value of x?

A. -1 B. 0 C. 1 D. 2 E. None of these

Question Time

```
x = 1
if x > 0:
    x = x - 2
if x < 0:
    x = x + 3
print (x)
```

What is the output?

- A. 2 B. -1 C. 4 D. None of these

Question Time

```
x = 1
if x==2 or 3 or 4:
    print ('yes')
else:
    print ('no')
```

What is the output?

- A. 'yes'
- B. 'no'
- C. 'The Denver Broncos'
- D. An error message

Question Time

```
x = 'EWNNESE'  
n = len(x)  
if x[n-1:n]=='ES':  
    print ('South East')  
elif x[n-2:n-1]=='NE':  
    print ('North East')  
else:  
    print (x[n/2])
```

- Output?
- (A) South East
 - (B) North East
 - (C) An error message is printed
 - (D) None of these

Question Time

```
def f(x):  
    z = 2*x;  
    y = z+1;  
    return y  
if __name__ == '__main__':  
    z = 10;  
    x = f(4)  
    print (z,x)
```

What is the output?

- A. 10 4 B. 10 9 C. 8 4 D. 8 9

Question Time

```
def f(x) :  
    y = 2*x  
    print (y)  
  
if __name__ == '__main__':  
    f(4)  
    z = f(4)  
    print (z)
```

What is the output:

- | | | | | | | | |
|----|---|----|------|----|------|----|---|
| A: | 8 | B: | 8 | C: | 8 | D: | 8 |
| | 8 | | 8 | | None | | |
| | | | None | | 8 | | |

Question Time

```
>>> s1 = input('First String: ')
>>> n1 = s1.count('ab')
>>> s2 = input('Next String: ')
>>> n2 = s2.count('ab')
>>> s = s1 + s2
>>> B = n1+n2 == s.count('ab')
```

What can you say about the value of B?

- A. Always True
- B. Always False
- C. Can be either True or False

Question Time

```
>>> s = 'abcabcabc'
>>> s.find('ca')
2
>>> n = s.find('bc')+s.find('bc')
>>> print (n)
```



What is the green box?


A. 2

B. 4

C. 7

Question Time

```
>>> s = 'abcdef'
>>> s.replace('bc', 'xx')
'axxdef'
>>> u = s.replace('de', 'yy')
>>> print (u)
```



What is the green box?

A. 'axxdef' B. 'abcyf' C. 'axxyf'

Question Time

```
s = '12345'  
t = 'x'  
for c in s:  
    t = t+t  
print (len(t))
```

Output?

A. 10 B. 15 C. 32 D. None of These

Question Time

```
T = ''  
S = 'abccabccabc'  
for c in S:  
    if T.count(c)==0:  
        T = T + c  
print (T)
```

Output?

- A. 'ccc' B. 'abc' C. 'cba'
D. None of These

String manipulation 1

- Implement the following function so that it performs as specified.

```
def Q1(s):
```

```
    """ Returns True if the characters at the start and  
    end of s are the same and occur nowhere else in s
```

```
    PreCondition: s is a string with length greater  
    than or equal to 3. """
```

String manipulation 2

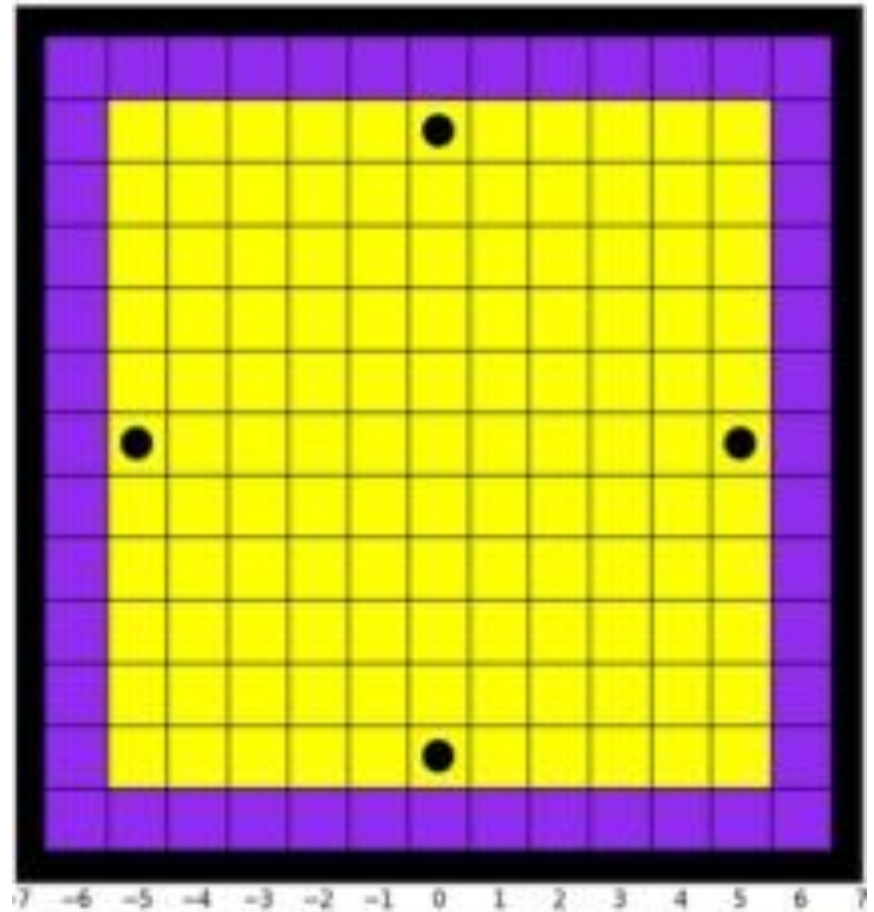
- Imagine playing around with this script:

```
s = input('Enter a string that has length greater than or equal to 2: ')
t = s.replace(s[0], 'x')
u = t.replace('x', s[0])
print (s, u)
```

- Sometimes it is the case that the printed values of `s` and `u` are the same and sometimes it is observed that they are different. Give a Boolean expression that is `True` if `u` and `s` have the same value and is `False` otherwise. Hint. Consider some small examples.

Random walk

- A random walk simulation produces a travel string comprised of the characters N, S, E, and W. The travel string encodes the hop directions associated with the robots journey from (0,0) to a purple boundary tile. Here is a display of an $n = 5$ “hopping arena” highlighting its four middle edge tiles (solid black dots):



Random walk 1

- Assume that x and y are initialized with the (x,y) coordinates of the robot's location and that the value of n is the size of the hopping arena. Give a Boolean expression that is True if the robot is on a middle edge tile and False otherwise.

Random walk 2

- A hop is “predictable” if it is in the same direction as the previous hop. Here is a travel string that includes 3 predictable hops: 'EWNNNWWN'. Complete the following function so that it performs as specified.

```
def nPredictable(s):  
    """ Returns an int that is the number of  
    predictable hops in s. Precondition: s is  
    a travel string.  
    """
```


Q1

- Assign a value to x so that the character 'A' is printed out:

x = _____

```
if x%2==0 and x%3==1:  
    print ('A')
```

Q2

- Assign values to x and y so that the character 'D' is printed out:

x = _____

y = _____

```
if not ((0<=x<=3) and (0<=y<=3)):
```

```
    print ('A')
```

```
elif y<=1 or y>=2:
```

```
    print ('B')
```

```
elif x<=1 or x>=2:
```

```
    print ('C')
```

```
else:
```

```
    print ('D')
```

Q3

- What would be the output if the following code is executed?

```
x = float(10/4)  
print(x)
```

Q4

- Suppose the functions in modules M1.py and M2.py are to be used by module M.py. Briefly explain why it is safer to implement M.py with

```
import M1  
import M2
```

- As opposed to with

```
Import M1 *  
Import M2 *
```

Q5

- Indicate what the output would be if the following application script is run:

```
def F(x,y):  
    x = y  
    y = x  
    z = x+2*y  
    print x,y,z  
    return z
```

```
if __name__ == '__main__':  
    x = 1  
    y = 2  
    print x,y  
    x = F(y,x)  
    print x,y  
    if x<y:  
        print 'A'  
    else:  
        print 'B'
```

Loops 1

- Consider the following script

```
t = 'x'
```

```
s = input('Enter a string: ')
```

```
for c in s:
```

```
    t = t + c + t
```

- Assuming that 'ba' is assigned to s, what is the final value of t? Show work.

Loops 2

- Write a script that is equivalent to the following script but which uses a while-loop instead of a for-loop.

```
t = 'x'
```

```
s = input('Enter a string: ')
```

```
for c in s:
```

```
    t = t + c + t
```