You can find multiple choice review questions (when they are available) in D2L under:
`Assessments->Quizzes`

For all questions, unless otherwise specified assume that there are no syntax errors in any programs or program fragments.

Short answer (code writing):

Begin with the starting program. It creates a variable sized rectangular list with each element initialized to a space.

```
MAX = 16
EMPTY = " "
CRITTER = "*"

def create():
    r = -1
    c = -1
    numRows = -1
    numColumns = -1

    aList = []
    numRows = random.randrange(0,MAX)+5
    numColumns = random.randrange(0,MAX)+5
    r = 0

    while(r < numRows):
        c = 0
        aList.append([])
        while(c < numColumns):
            aList[r].append(EMPTY)
            c = c + 1
        c = 0
        #while(c < numColumns):
        #    print(ord(aList[r][c]), end = "-")
        #    c = c + 1
        #print()
        r = r + 1
    return(aList,numRows,numColumns)
```

```
def display(aList,numRows,numColumns):
    r = -1
    c = -1
    r = 0
    while(r < numRows):
        c = 0
        while(c < numColumns):
            print(" -" , end = "")
            c = c + 1
        print()

        c = 0
        while(c < numColumns):
            print("|%s" %(aList[r][c]), end = "")
            c = c + 1
        print("|")
        r = r + 1
    c = 0
    while(c < numColumns):
        print(" -" , end = "")
        c = c + 1
    print()

def start():
    numRows = -1
    numColumns = -1
    rowToEdit = -1
    columnToEdit = -1
    aList = []
    aList,numRows,numColumns = create()
    print("List size (rows/columns)=(%d/%d)" %(numRows,numColumns))
    display(aList,numRows,numColumns)
    rowToEdit,columnToEdit = getLocationToEdit(numRows,numColumns)
    if(isInside(rowToEdit,columnToEdit,numRows,numColumns) == True):
        edit(aList,rowToEdit,columnToEdit)
    display(aList,numRows,numColumns)


start()
```

Write the code for the following functions:

`def getLocationToEdit(maxRows,maxColumns):`

- **Parameters:** `(int,int)` The arguments specify the size of the list which is randomly generated in the `create()` function.
- **Return value:** `(int,int)`: The row/column entered by the user.
- **Functionality:** Prompts for two integers from the user. These integers are returned back to the caller.

`def isInside(row,column,maxRows,maxColumns):`

- **Parameters:** `(int,int,int,int)` The first two arguments specify a candidate location to be edited. The last two arguments specify the size of the list
- **Return value:** `(Boolean)`: Specifies if the location specified by the first pair of arguments is within the bounds of the list.
- **Functionality:** Prompts for two integers from the user. These integers are returned back to the caller.

`def edit(aList,row,column):`

- **Parameters:** `(list,int,int)`
- **Return value:** none
- **Functionality:** The (row/column) coordinate of a location (specified in the argument list) is changed to a 'Critter' (star). The responsibility for checking that the location is in bounds is performed in another function so there is no need to implement this again in the `edit()` function.