Northern Michigan University (Marquette Co, MI)

1. Where does onResume() fire in the grand scheme of the activity lifecycle? In what common situation might you want to put code in there as opposed to onCreate() or onStart()?

onResume() fires AFTER onStart(). In a nutshell, onResume() fires when the activity is ready to accept user interaction. (onStart() fires when the activity becomes visible to the user.) The most common use of onResume() is when a dialog box CLOSES. The app, already visible, was paused due to the presence of the dialog box. Once the box is closed, the app can resume. More generally, onResume() is the best place to put code for when the program is recovering from being paused, but not specifically from being stopped.

2. Write, in Kotlin, that portion of onCreate() in a *secondary* activity that identifies a button mButt in the Kotlin code with a specific button in the layout called butt1, and two EditTexts called in the code mET1 and mET2 and in the layout et1 and et2. When the button is pressed, whichever string in the two EditTexts come first in alphabetical order is returned to the main activity, ending this activity.

```
mButt = findViewById(R.id.butt1)
mET1 = findViewById(R.id.et1)
mET2 = findViewById(R.id.et2)
mButt!!.setOnClickListener {
    var answer:String = ""
    if (mET1!!.text.toString() < mET2!!.text.toString())
        answer = mET1!!.text.toString()
    else
        answer = mET2!!.text.toString()
    val data:Intent? = Intent()
    data!!.putExtra ("smallest",answer)
    setResult (RESULT_OK,data)
    finish()
}
```

3. I have the following method in Kotlin:

```
fun BigOlX (A:Array<String>) { ... }
```

Every string in A is the same length as every other and that length is the same number as the number of strings in A. You may trust that this is so; you do not have to verify it. In other words, you might envision A has a square grid of characters, the number of rows and the number of columns are the same. Write Kotlin code that puts a diagonal of X's from the upper left to the lower right, and another from the upper right to the lower left. All other characters should be underscores. For example, if A had five strings (each of length 5) you should put the following into A:

```
X___X
_X_X_
__X__
_X_X_
X___X
```

Write the code for this method. Be careful not to go out of bounds.

```
fun BigOlX (A:Array<String>) {

 for (i in 0..A.size-1)
  for (j in 0..A[i].length-1)
   A[i][j] = '_'
 for (i in 0..A.size-1) {
  A[i][i] = 'X'
  A[i][A.size-i-1] = 'X'
}
```

4. An SQLite database named "cars.db" contains a table called "CARS". CARS contains a field called "MAKE", a field called "MODEL", and a field called "YEAR". Write Kotlin code that prints (to the screen, not to the android) the makes and models of all cars made in the 1970's. You do not have to write the code to open the database. You can assume it's already open and assigned to an SQLite? variable named mCars. You may also assume that the query method and the wrapper class has already been written.

```
var cursor = query ("YEAR >= 1970 and YEAR <= 1979",null)
 cursor.moveToFirst()
 while (!cursor.isAfterLast){
  val carinfo = cursor.stuff
  println (
"20TH CENTURY:  ${carinfo.MAKE} ${carinfo.MODEL}")
  cursor.moveToNext()
 }
```

5. Write the Kotlin code that will open the image file ("http://philos.nmu.edu/floralarrangement.jpg") and display it on a ImageView named PrettyFlowers.

```
mPrettyFlowers = findViewById (R.id.prettyflowers)
val executor = Executors.newSingleThreadExecutor()
executor.execute {
 doInBackground("http://philos.nmu.edu/floralarrangement.jpg")
}
while (!webthread);
onPostExecute (mIcon11!!,mPrettyFlowers!!)


fun doInBackground(url: String)  {

 val infile = java.net.URL(url).openStream()
 mIcon11 = BitmapFactory.decodeStream(infile)
 webthread = true
}

fun onPostExecute (result:Bitmap,pfImage:ImageView) {
```

```
 pfImage.setImageBitmap(result)
}
```

6. Write the Kotlin code that will play the first 15 minutes and 12 seconds of Beethoven's Fifth Symphony. This song can be found at http://philos.nmu.edu/dadadadum.mp3 .

```
val mp = MediaPlayer()
mp.setDataSource ("http://philos.nmu.edu/dadadadum.mp3")
mp.prepare()
mp.start()
try {
  TimeUnit.SECONDS.sleep (912)
} catch (e:Exception) {
}
mp.pause ()
```