# Northern Michigan University (Marquette Co, MI)

Time:  50 minutes.

1.  Write the following method in Kotlin:

fun WeirdSum (A:Array<Int>):Int { ... }

This method computes the sum of the integers in the array, except that it squares the even numbers and cubes the odd ones before summing.  For example, if the array were {1,2,3}, the method would return 32.

```
var sum = 0
for (i in A)
 if (i%2==0) sum += i*i
 else sum += i*i*i
return sum
```

2.  My main activity calls a secondary activity.  I can't get my "finish" code to work properly because I don't really understand it, so instead of having the secondary activity return control to the main activity, I just start a new main activity.

What's wrong with this way?  What problems could this cause?

**If you create new activities instead of returning to the old, the older activities take up space on the device but contribute nothing to the app.  If this happens to many times in the app, it will crash.  Additionally, any work done by the first main activity is lost unless it is passed in intents.  So, it uses space needlessly and complicates the process of storing and retrieving information.**

3.  Given the Kotlin class definitions below for a linked list and a linked list node:

```
class LL {                              class LLN (w:String,n:LLN?) {

 var head:LLN? = null                    var word = w
}                                        var next = n
                                        }
```

Write code for the LL method `fun first():LLN?` .  This method returns a pointer to the node that contains the first string in alphabetical order.  (If there is a tie for first, I don't care which one you return.)  You may add methods to LL and to LLN as you see fit.  Remember:  you are returning a pointer to a node.  You are not returning the string itself.

```
LL:
fun first():LLN? {

 return head?.first()
}
```

# Northern Michigan University (Marquette Co, MI)

CS345-01-23W Android Programming (Andrew A. Poe)   Name: _____
Practice Midterm Examination (Exam 1) Page 2/2                    Monday 27 February 2023 1:00 P.M. EST

```
LLN:
fun first():LLN? {

 val best=next?.first()
 if (best==null || word <= best!!.word) return this
 return best
}
```

4.  We have a button (myButt) and a text view (myTV) that have already been connected to widgets (so you don't have to do that). Write the code so that when the button is clicked. the text view is examined.  If the text view contains "A", a new activity (AActivity) is started.  If the text view contains "B", a different new activity (BActivity) is started.  No parameters need to be added to the intent in either case.

```
myButt!!.setOnClickListener {

 var i:Intent? = null
 if (myTV!!.text!!.toString()!!=="A")
  i = AActivity.newIntent (this)
 else if (myTV!!.text!!.toString()!!=="B")
  i = BActivity.newIntent (this)
 startActivity (i)
}

AActivity:

fun newIntent (packageContext:Context?) {

 return Intent (packageContext!!,AActivity::class.java)
}

BActivity:

fun newIntent (packageContext:Context?) {

 return Intent (packageContext!!,BActivity::class.java)
}
```