CISS245: Advanced Programming
Quiz q06

Name: jdoe5@cougars.ccis.edu Score:

Open `main.tex` and enter answers (look for `answercode`, `answerbox`, `answerlong`). Turn the page for detailed instructions. To rebuild and view pdf, in bash shell execute `make`. To build a gzip-tar file, in bash shell execute `make s` and you'll get `submit.tar.gz`.

Q1. Write function
`void merge(int x[], int & x_len , int y[], int y_len, int z[], int z_len)`
where `y[0]`, `y[1]`, ... `y[y_len - 1]` and `z[0]`, `z[1]`, ... `z[z_len - 1]` are both sorted and the values in array `y` and `z` are "merged" into `x` so that `x` is sorted and `x_len` is set to `y_len + z_len`. For instance if `y` is `{0, 2, 6}` (and `y_len` is 3) and `z` is `{0, 1, 2, 5, 7, 9}` (and `z_len` is 5), then `x` becomes `{0, 0, 1, 2, 2, 5, 6, 7}` and `x_len` is 8. Iteratively, the least among what is left in `y` and `z` is selected to be placed in `x`.

```
y = {0, 2, 6}, z = {0, 1, 2, 5, 7, 9}, x = {0, ?, ?, ?, ?, ?, ?, ?, ?}
y = {0, 2, 6}, z = {0, 1, 2, 5, 7, 9}, x = {0, 0, ?, ?, ?, ?, ?, ?, ?}
y = {0, 2, 6}, z = {0, 1, 2, 5, 7, 9}, x = {0, 0, 1, ?, ?, ?, ?, ?, ?}
y = {0, 2, 6}, z = {0, 1, 2, 5, 7, 9}, x = {0, 0, 1, 2, ?, ?, ?, ?, ?}
y = {0, 2, 6}, z = {0, 1, 2, 5, 7, 9}, x = {0, 0, 1, 2, 2, ?, ?, ?, ?}
y = {0, 2, 6}, z = {0, 1, 2, 5, 7, 9}, x = {0, 0, 1, 2, 2, 5, ?, ?, ?}
y = {0, 2, 6}, z = {0, 1, 2, 5, 7, 9}, x = {0, 0, 1, 2, 2, 5, 6, ?, ?}
y = {0, 2, 6}, z = {0, 1, 2, 5, 7, 9}, x = {0, 0, 1, 2, 2, 5, 6, 7, ?}
y = {0, 2, 6}, z = {0, 1, 2, 5, 7, 9}, x = {0, 0, 1, 2, 2, 5, 6, 7, 9}
```

Note: there must be no sorting done on `x`.
Answer:

```
void merge(int x[], int & x_len , int y[], int y_len, int z[], int z_len)
{
}
```

(Hint on next page if needed.)

Hint

Use 3 index variables, one for each array. Here's a pseudocode for you:

```
i = 0 // index for x
j = 0 // index for y
k = 0 // index for z

while j < y_len and k < z_len:
    compare y[j] and z[k]
    put the smaller of the two into x[i] and increment i
    increment j if y[j] was used, otherwise increment k

while j < y_len:
    copy y[j] to x[i] and increment i and j

while k < z_len:
    copy z[k] to x[i] and increment i and k

set x_len appropriately (using i)
```