

# Northern Michigan University (Marquette Co, MI)

CS444-01-25F: Parallel & Distributed Processing (Andrew A. Poe)  
Quiz 2

Name: \_\_\_\_\_  
Wednesday 10 September 2025 9:00 A.M. EDT

I have three variables declared in global memory:

```
int *orig, *lrg, sz;
```

orig and lrg are arrays of int of length sz. They are initialized in a main method (which you do not have to write).

Write the method void LargestHereDown () (in C++) which builds the lrg array, by making each entry in the lrg array equal to the largest element in the portion of the orig array starting with the corresponding position to the current position in the lrg array and ending with the last position in the orig array.

For example, if the original array were {1,9,2,8,3,7}, the lrg array would become {9,9,8,8,7,7}.  
The largest entry from the 1 to the 7 is the 9.  
The largest entry from the 9 to the 7 is the 9.  
The largest entry from the 2 to the 7 is the 8.  
And so forth.

Each entry in the sum array should have a thread dedicated to computing its value.

```
void Lrg () {  
  
    int *id = new int[sz];  
    th = new pthread_t[sz];  
    for (int i=0; i < sz; i++) {  
        id[i] = i;  
        pthread_create (&th[i],nullptr,getlrg,&id[i]);  
    }  
    for (int i=0; i < sz; i++)  
        pthread_join (th[i],nullptr);  
    delete[] id;  
    delete[] th;  
}
```

```
void * getlrg (void *a) {  
  
    int idnum = *(int *)a;  
    lrg[idnum] = orig[idnum];  
    for (int i=idnum; i < sz; i++)  
        if (orig[i] > lrg[idnum])  
            lrg[idnum] = orig[i];  
    return nullptr;  
}
```