Lab #10: Review Exam Codes

Getting started

Download lab10 materials from D2L (including this instruction and three starter codes)
Enter Mimir IDE.
Change into the cse220 directory.
Create a new directory called lab10.
Change into the new directory.
Upload the starter codes to Mimir IDE, save it in /home/(your_username)/cse220/lab10/
Implement the programs below in your lab10 directory.

Question 2: No indexing

Read and rewrite the following function `find_max`, which finds the maximum value from the `n` integer elements of the array `a`, with pointers. No square brackets [] should appear within your codes.

```
int find_max(int a[], int n)

{

      int max = a[0];

      for (int i = 1; i < n; i++) {

             if (a[i] > max) {

                 max = a[i];

              }

       }

       return max;

}
```

We provide a starter code (which includes an example input {6,2,7,1}) to help you compile and debug your code. If your code is right, it should be able to output the following result:

```
user@mimir: ~ > gcc -o Q2 Q2.c
user@mimir: ~ > ./Q2
The maximum integer is: 7
```

Question 4: String

Write the function `strlow` in order to change all the letters of string `a` to the corresponding lower cases. `a` has `n` elements.

```
void strlow(char a[], int n)
```

For example, if a[] = "AbcDe", n = 5, the result would be "abcde".
We provide a starter code (which includes the above example input) to help you compile and debug your code. If your code is right, it should be able to output the following result:

```
user@mimir: ~/cse220/lab10 > gcc -o Q4 Q4.c
user@mimir: ~/cse220/lab10 > ./Q4
abcde
```

Question 5: Function

Write the following function `squared_euclidean` to calculate the **squared Euclidean distance** of two arrays `a` and `b`, of which the size is `n`.

```
int squared_euclidean(int a[], int b[], int n)
```

If the first array `a` is $\{a_1, a_2, a_3, ..., a_n\}$, the second array `b` is $\{b_1, b_2, b_3, ..., b_n\}$, then their squared Euclidean distance is:

$$(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + ... + (a_n - b_n)^2$$

For example, if `a = {1, 2, 3}`, `b = {4, 5, 6}`, their squared Euclidean distance would be:

$$(1 - 4)^2 + (2 - 5)^2 + (3 - 6)^2 = 9 + 9 + 9 = 27$$

We provide a starter code (which includes the above example input) to help you compile and debug your code. If your code is right, it should be able to output the following result:

```
user@mimir: ~ > gcc -o Q5 Q5.c
user@mimir: ~ > ./Q5
27
```