

Homework 6
Due 5pm, Wednesday, May 17, 2017

Download the starter code `main.cpp`, `hw6.h`, and `hw6.cpp`. You will only turn in `hw6.cpp`. Do not modify `hw6.h`, as you have no reason to. We have provided `main.cpp` to give you an idea of how we intend to use the functions. You are free to do whatever you want with `main.cpp` (and you'll want to make changes to it for debugging) but you won't submit it. `hw6.cpp` must not contain a `main` function. You may not use global variables.

We may take off up to 20% of the total marks for poor style; make sure to name your variables reasonably, indent properly, and comment sufficiently. Submit `hw6.cpp`.

Problem 1: (Integer power)

Write the implementation of

```
int ipow(int base, int exp);
```

which returns `base` to the `exp` power. Assume `exp` is nonnegative. Note that $0^0 = 1$. You may not use any libraries aside from `cassert`.

Remark. If you were to use the `cmath` library, you could do

```
int ipow(int base, int exp) {  
    return static_cast<int>(pow(double(base), double(exp)));  
}
```

Remark. You could consider `exp==0` an edge case. Sometimes edge cases must be handled separately with, say, an `if`-statement. However, you can sometimes avoid additional control structures by being mindful of how you write your loops and how you initialize your variables. Avoid additional control structures if you can.

Problem 2: (Minimum)

Write the implementation of

```
double my_min(double* arr, int len);
```

which returns the minimum value within `arr`, an array of length `len`. You may not use any libraries aside from `cassert`.

Problem 3: (C-style strings)

Read the documentation for `strcpy` and `strcmp`

<http://www.cplusplus.com/reference/cstring/strcpy/>

<http://www.cplusplus.com/reference/cstring/strcmp/>

Write the functions

```
char* my_strcpy(char* destination, const char* source);
```

and

```
int my_strcmp(const char* str1, const char* str2);
```

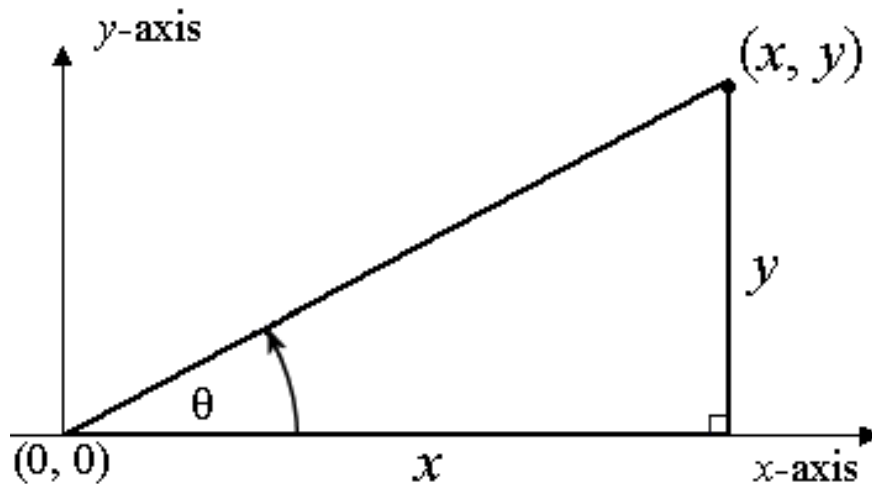
so that they do exactly same thing as `strcpy` and `strcmp`, respectively. You may not use any libraries aside from `cassert`.

Problem 4: (Angle)

Write the implementation of

```
double angle(double x, double y);
```

The function returns the angle θ between the line segment connecting the origin and the point (x,y) and the x-axis of the xy-plane. The figure should be clear. The output is in degrees and is between 0, inclusive, and 360, exclusive. You may not use any libraries aside from `cmath` and `cassert`. You may not use the function `atan2` of the `cmath` library.



Clarifications.

- `angle(1,0)` returns 0.
- `angle(1,1)` returns 45.
- `angle(0,1)` returns 90.
- `angle(-2,2)` returns 135.
- `angle(-2,0)` returns 180.
- `angle(-1,-1)` returns 225.
- `angle(0,-1)` returns 270.
- `angle(1,-1)` returns 315.
- `angle(1,-0.00001)` returns something like 359.999.

Hint. Make sure you handle all edge cases.

Hint. If you don't think carefully,

```
double angle(double x, double y) {  
    return atan(y/x);  
}
```

might seem like a correct answer. This is a good starting point but not a final answer.

Hint. Read the documentation of `atan` at <http://www.cplusplus.com/reference/cmath/atan/>