Test 2

Question 1:

```
Test with two integers
Min of 1 and 4: 1
Min of 4 and 1: 1

Test with two floats
Min of -12 and 400.6: -12
Min of 400.6 and -12: -12

Test with two doubles
Min of 100 and 40.6: 40.6
Min of 40.6 and 100: 40.6
```

Question 2:

```
Thank you for joining 'CCET4690', we hope you have a good semester

PS F:\test2>
```

Question 3:

Explain the purpose of the Time class in your code.

- The purpose of the time class is to keep track of a specific hour, minute and second. It provides a convenient way to hold and display times.

Describe the role of the constructor Time::Time(int hour, int minute, int second).

- The constructor sets the initial time based on the input variables. It's called when a new object is created.

What is the significance of the setTime(int hour, int minute, int second) function within the Time class?

- This function updates the private variables with a new hour, minute and second.

Explain how the printUniversal() function prints the time in the universal-time format.

- For the hour, minute and second this function checks to see if the digit needs to be padded and then outputs the digit.

Highlight the difference between printUniversal() and printStandard() functions in terms of the time format they produce.

- The difference between the two is in the way the hour is displayed. For the printStandard function, if the hour is greater than 12 than 12 is subtracted from the hour and PM is outputted instead of AM.

```
The current standard time: 01:02:03 AM

The current standard time: 08:05:06 PM
The current universal time: 20:05:06

The current universal time: 23:59:59
```

Test 2

Question 4:

```
Testing basic matrix functionality
------------------
Matrix length: 16
Matrix:
[ 0 0 0 0 ]
[ 0 0 0 0 ]
[ 0 0 0 0 ]
[ 0 0 0 0 ]

Setting each value:
Matrix:
[ 0 1 2 3 ]
[ 4 5 6 7 ]
[ 8 9 10 11 ]
[ 12 13 14 15 ]

Getting each value:
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Testing addition
------------------
Setting each value:
[ 0 2 4 6 ]
[ 8 10 12 14 ]
[ 16 18 20 22 ]
[ 24 26 28 30 ]

+
[ 1 3 5 7 ]
[ 9 11 13 15 ]
[ 17 19 21 23 ]
[ 25 27 29 31 ]

=
[ 1 5 9 13 ]
[ 17 21 25 29 ]
[ 33 37 41 45 ]
[ 49 53 57 61 ]

Testing subtraction
------------------
[ 0 2 4 6 ]
[ 8 10 12 14 ]
[ 16 18 20 22 ]
[ 24 26 28 30 ]

-
[ 1 3 5 7 ]
[ 9 11 13 15 ]
[ 17 19 21 23 ]
[ 25 27 29 31 ]

=
[ -1 -1 -1 -1 ]
[ -1 -1 -1 -1 ]
[ -1 -1 -1 -1 ]
[ -1 -1 -1 -1 ]

Testing multiplcation
------------------
Setting each value:
[ 0 8 16 ]
[ 16 24 18 ]

*
[ 1 3 ]
[ 9 11 ]
[ 17 19 ]

=
[ 344 392 ]
[ 538 654 ]
```