Group Members: Aidan Murphy ,Bobby Hao, Jay Ahlschwede, Israel Prado

Top Down Design Hierarchy

Create ASCII Clock

Get Input

Format Input

Split string into list

Create character key

Print formatted time

Lists of Lists for numbers

Print statement with formatted number

Print Invalid for invalid times

If statements for impossible times

Assignments: Bobby,Israel,Aidan,Jay

Variables

a,b,c,d,e ##:Correspond to each character in the ASCII format to be printed
tin: string format of an inputted time
diglist: a list of digits that are in the time
time1: first character of the time
time2: second character of the time
time3: third character of the time
time4: fourth character of the time

Test Cases

1. (1:00) Typical, Expected Output: 1:00 in ASCII
2. (2:44) Typical, Expected Output: 2:44 in ASCII
3. (12:59) Edge, Expected Output: 12:59 in ASCII
4. (3:59) Edge, Expected Output: 3:59 in ASCII
5. (4:60) Corner,Expected Output: Invalid time

Questions

Q1: The difficulty of combining code was moderate, we tried to be specific with where to start/stop so that helped communication. It was challenging to compare variables, but once those were consistent it was more smooth. We also had a team member focused on integration as well. This left us with ideas of how to create separate parts that fit together better. One way would be to use a consistent key for general use variables, and work more dynamically as we create our code.

Q2: A major benefit is a lighter workload for each team member, which is always beneficial. This also allows for each piece of code to take a higher quality because the member working on it is focused on just it. This tends to lead to a more refined code project then integrating four solo projects. A drawback is that the initial plan has to work, because just

creating your piece of code doesn’t guarantee a working, finished project. This method also seems to fail without good communication and teamwork.