

Purpose: Using control structures

For this lab you will write a small program that will prompt the user for three numbers representing the time as hours, minutes, and seconds. Your program will then print the time as a series of asterisks (\*), with one digit represented on each line (B and C grades) or on two lines (A grade). After printing the asterisks, the code will loop back to input and print a new time unless the hours, minutes and seconds are all zero. In that case the program will exit.

For the C grade print each value as simply that number of asterisks (e.g., 5 is printed as "\*\*\*\*\*").

For the B grade print each value in 6-bit binary with underscore ('\_') for a zero, and asterisk for a one (e.g., 5 is printed as "\_\*\_\*\_\*").

For the A grade print each value in BCD using underscores and asterisks. (e.g, 35 would print on two lines: "\_\*\_\*\_\*" and "\_\*\_\*\_\*").

You must complete the lower grade before moving on to a higher grade. Sample output for each grade is given below. Your code should produce identical results to this.

When finished, have a TA check your solution.

Notes and hints:

- 1) Your program should contain an outer loop that manages prompting for the time and then printing each line. This loop will then test to see if the three values are all zero and it will exit if so. Use the appropriate control structure for this.
- 2) You should print asterisks for the three values using an inner loop: the first time through the loop the hours are printed, the second time the minutes are printed and the last time, the seconds are printed. To do this you will need to declare a number called `value`, and at the top of the loop set this value to either the hours, the minutes, or the seconds, depending on which time through the loop this is.
- 3) To print the asterisks (for any grade), you will also need a loop. Note we now have a loop within a loop within a loop.
- 4) Check each value before printing the line of asterisks for it. If the value is not between 0 and 59, inclusive, then print "ERROR: <value>" instead of that line, where "<value>" is the value.
- 5) Hint: for the B grade, you will need to determine the bits of the binary representation of each number. You can test each bit by ANDing the number with a mask that has a one in the appropriate position. Shift the mask as you go on to the next bit.
- 6) Hint: for the A grade, you will need to convert each value to BCD. The 10's digit is simply the value divided by 10, and the one's digit is simply the value modulo 10.

Sample output from your program for each of the three grades:

[C grade]

```
Enter hours, min, sec: 12 34 56
*****
*****
*****
```

```
Enter hours, min, sec: 3 -4 5
***
ERROR: -4
*****
```

```
Enter hours, min, sec: 0 0 0
```

[B grade]

```
Enter hours, min, sec: 12 34 56
  **
_  _
*  *
*** _
```

```
Enter hours, min, sec: 3 -4 5
  **
_  _
ERROR: -4
  * *
```

```
Enter hours, min, sec: 0 0 0
_____
_____
_____
```

[A grade]

```
Enter hours, min, sec: 12 34 56
  *
_  *
_  **
_  *
_  * *
_  **
_  _
```

```
Enter hours, min, sec: 3 -4 5
  **
_  _
ERROR: -4
  * *
```

```
Enter hours, min, sec: 0 0 0
_____
_____
_____
_____
_____
```