Due: Wednesday, October 23rd. Written: 4pm in 2131 Kemper. Programs: 11:59pm using handin to cs30 p3 directory. Filenames: weight.c, quadrant.c, emissions.c, wireless.c, bread.c, baseball.c

Written (2 points): pp. 229-230: 5, 6

#5. Write an if statement that displays an acceptance message for an astronaut candidate if the person’s weight is between the values of opt_min and opt_max inclusive, the person’s age is between age_min and age_max inclusive, and the person is a nonsmoker (smoker is false).

#6 Implement the flow diagram in Fig. 4.14 using a nested if structure.

**Programming (45 points)**

All programs should be able to compile with no warnings when compiled with the –Wall option. You should put your name(s) in a comment on the first line of each file. The prompts, and output format of each program must match the examples exactly. You will find my executables in ~ssdavis/30/p3 in the CSIF. Remember to a blank line before and after each unindented if statement.

1. p. 231 #2 (5 points, 5 minutes) Filename: weight.c
   “Write a program that calculates the user’s body mass index (BMI) and categorizes it as underweight, normal, overweight, or obese, based on the table from the United States Centers for Disease Control:

<table>
<thead>
<tr>
<th>BMI</th>
<th>Weight Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below 18.5</td>
<td>Underweight</td>
</tr>
<tr>
<td>18.5 – 24.9</td>
<td>Normal</td>
</tr>
<tr>
<td>25.0 – 29.9</td>
<td>Overweight</td>
</tr>
<tr>
<td>30.0 and above</td>
<td>Obese</td>
</tr>
</tbody>
</table>

   To calculate BMI based on weight in pounds (wt_lb) and height in inches (ht_in), use this formula (rounded to tenths):
   \[ BMI = \frac{703 \times wt_{lb}}{ht_{in}^2} \]
   Prompt the user to enter weight in pounds and height in inches.”

   ssdavis@lect1 p3]$ weight.out
   Please enter your weight in whole pounds: 110
   Please enter your height in whole inches: 60
   You have a BMI of 21.5, and your weight status is normal.
   [ssdavis@lect1 p3]$ weight.out
   Please enter your weight in whole pounds: 95
   Please enter your height in whole inches: 61
   You have a BMI of 17.9, and your weight status is underweight.
   [ssdavis@lect1 p3]$ weight.out
   Please enter your weight in whole pounds: 180
   Please enter your height in whole inches: 69
   You have a BMI of 26.6, and your weight status is overweight.
   [ssdavis@lect1 p3]$ weight.out
   Please enter your weight in whole pounds: 223
   Please enter your height in whole inches: 70
   You have a BMI of 32.0, and your weight status is obese.
   [ssdavis@lect1 p3]$

2. p. 232 #6 (5 points, 8 minutes) Filename quadrant.c
   “Write a program that takes the x-y coordinates of a point in the Cartesian plane and prints a message telling either an axis on which the point lies of the quadrant in which it is found.”
Please enter the x and y coordinates: -1.0 -2.5
(-1.0, -2.5) is in quadrant III

Please enter the x and y coordinates: 0.0 4.8
(0.0, 4.8) is on the y axis

Please enter the x and y coordinates: 43.2 57.1
(43.2, 57.1) is in quadrant I

Please enter the x and y coordinates: 93.5 0.0
(93.5, 0.0) is on the x axis

Please enter the x and y coordinates: 0.0 0.0
(0.0, 0.0) is the origin

Please enter the x and y coordinates: -33 47
(-33.0, 47.0) is in quadrant II

Please enter the x and y coordinates: 3.9 -12
(3.9, -12.0) is in quadrant IV

3. p. 233 #8 (5 points, 7 minutes) Filename: emissions.c

“Write a program that interacts with the user like this:

(1) Carbon monoxide
(2) Hydrocarbons
(3) Nitrogen oxides
(4) Nonmethane hydrocarbons

Enter pollutant number>> 2
Enter number of grams emitted per mile>> 0.35
Enter odometer reading>> 40112

Emissions exceed permitted level of 0.31 grams/mile.

Use the table of emissions limits below to determine the appropriate message.1

<table>
<thead>
<tr>
<th>Pollutant Type</th>
<th>First 50,000 Miles</th>
<th>Second 50,000 Miles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbon monoxide</td>
<td>3.4 grams/mile</td>
<td>4.2 grams/mile</td>
</tr>
<tr>
<td>Hydrocarbons</td>
<td>0.31 grams/mile</td>
<td>0.39 grams/mile</td>
</tr>
<tr>
<td>Nitrogen oxides</td>
<td>0.4 grams/mile</td>
<td>0.5 grams/mile</td>
</tr>
<tr>
<td>Nonmethane hydrocarbons</td>
<td>0.25 grams/mile</td>
<td>0.31 grams/mile</td>
</tr>
</tbody>
</table>


4. p. 233 #9 (10 points, 17 minutes) Filename: wireless.c

“Chatflow Wireless offers customers 600 weekday minutes for a flat rate of 39.99. Night (8pm to 7am) and weekend minutes are free, but additional weekday minutes cost 0.40 each. There are taxes of 5.25% on all charges. Write a program that prompts the user to enter the number of weekday minutes, night minutes, and weekend minutes used, and calculates the monthly bill and average cost of a minute before taxes. The program should display the taxes, and the total bill. Store all monetary values as whole cents (rounding the taxes and average minute cost), and divide by 100 for display of results.”
Additional specification: You must use constant macros for all constants given in the above paragraph. You should store 39.99 and 0.40 as cents rather than dollars. Note that since you are dividing by 100, and not 100.0, you will have to cast your int variables to doubles in your printf statements before dividing by CENTS_PER_DOLLAR. Read more about casting on pp. 76-77.

Since you have been asked to round the taxes and average cost, and converting from doubles to ints simply truncates, you will need to add 0.5 to produce the desired integer value, where x is a double.

```
[ssdavis@lect1 p3]$ wireless.out
Please enter the number of weekday minutes, night minutes: 600 50
Please enter the number of weekend minutes: 150
Weekday minutes: 600, night minutes: 50, weekend minutes: 150
Average minute cost: $0.05
Pretax bill: $ 39.99
Taxes:       $  2.10
Total bill:  $ 42.09
[ssdavis@lect1 p3]$ wireless.out
Please enter the number of weekday minutes, night minutes: 2348 1221
Please enter the number of weekend minutes: 348
Weekday minutes: 2348, night minutes: 1221, weekend minutes: 348
Average minute cost: $0.19
Pretax bill: $739.19
Taxes:       $ 38.81
Total bill:  $778.00
[ssdavis@lect1 p3]$ wireless.out
Please enter the number of weekday minutes, night minutes: 1328 47
Please enter the number of weekend minutes: 147
Weekday minutes: 1328, night minutes: 47, weekend minutes: 147
Average minute cost: $0.22
Pretax bill: $331.19
Taxes:       $ 17.39
Total bill:  $348.58
[ssdavis@lect1 p3]$
```

5. pp. 233-234 #10 (15 points, 25 minutes) Filename bread.c

"Write a program to control a bread machine. Allow the user to input the type of bread as W for White, and S for Sweet. Ask the user if the loaf size is double, and if the baking is manual. The following table details the time chart for the machine for each bread type. Display a statement for each step. If the loaf size is double, increase the baking time by 50 percent. If baking is manual, stop after the loaf-shaping cycle and instruct the user to remove the dough for manual baking. Use functions to display instructions to the user and to compute the baking time."

<table>
<thead>
<tr>
<th>Bread Time Chart</th>
<th>White Bread</th>
<th>Sweet Bread</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary kneading</td>
<td>15 mins</td>
<td>20 mins</td>
</tr>
<tr>
<td>Primary rising</td>
<td>60 mins</td>
<td>60 mins</td>
</tr>
<tr>
<td>Secondary kneading</td>
<td>18 mins</td>
<td>33 mins</td>
</tr>
<tr>
<td>Secondary rising</td>
<td>20 mins</td>
<td>30 mins</td>
</tr>
<tr>
<td>Loaf shaping</td>
<td>2 seconds</td>
<td>2 seconds</td>
</tr>
<tr>
<td>Final rising</td>
<td>75 mins</td>
<td>75 mins</td>
</tr>
<tr>
<td>Baking</td>
<td>45 mins</td>
<td>35 mins</td>
</tr>
<tr>
<td>Cooling</td>
<td>30 mins</td>
<td>30 mins</td>
</tr>
</tbody>
</table>

Additional specifications: All three input variables must be chars. The user may enter either capital or lowercase letters, but will not enter invalid letters. The display_instructions() function will NOT calculate the total baking time.
The calc_baking_time will compute and print the total baking time. You should read the section on Undetected Errors on pp. 95-98 before writing your scanf() calls. For debugging, I recommend that you add the line “printf("\%c \%c \%c\n", type, loaf_size, manual);” after your scanf() to ensure that their values are correct.

Extra credit (5 points): My calc_baking_time function uses only five “ifs”. If you can write a calc_baking_time function that uses five or fewer “ifs” (without using a switch statement), then you will earn five points extra credit.

```
[ssdavis@lect1 p3]$ bread.out
Are you making white or sweet bread (w or s)? w
Is this a single or double loaf (s or d)? S
Are you going to bake manually (y or n)? y
Primary kneading: 15 minutes
Primary rising: 60 minutes
Secondary kneading: 18 minutes
Secondary rising: 20 minutes
Loaf shaping: 2 seconds
You should remove the dough for manual baking.

For a total baking time of 113 minutes and 2 seconds.
[ssdavis@lect1 p3]$ bread.out
Are you making white or sweet bread (w or s)? S
Is this a single or double loaf (s or d)? d
Are you going to bake manually (y or n)? n
Primary kneading: 20 minutes
Primary rising: 60 minutes
Secondary kneading: 33 minutes
Secondary rising: 20 minutes
Loaf shaping: 2 seconds
Final rising: 75 minutes
Baking: 52 minutes 30 seconds
Cooling: 30 minutes

For a total baking time of 300 minutes and 32 seconds.
[ssdavis@lect1 p3]$ bread.out
Are you making white or sweet bread (w or s)? W
Is this a single or double loaf (s or d)? s
Are you going to bake manually (y or n)? N
Primary kneading: 15 minutes
Primary rising: 60 minutes
Secondary kneading: 18 minutes
Secondary rising: 20 minutes
Loaf shaping: 2 seconds
Final rising: 75 minutes
Baking: 45 minutes 0 seconds
Cooling: 30 minutes

For a total baking time of 263 minutes and 2 seconds.
[ssdavis@lect1 p3]$
```
6. p.290 #12 (7 minutes, 5 points) Filename baseball.c

“A baseball player’s batting average is calculated as the number of hits divided by the official number of at-bats. In calculating official at-bats, walks, sacrifices, and occasions when hit by the pitch are not counted. Write a program that takes an input file containing player numbers and batting records. Trips to the plate are coded in the batting record as follows: H—hit, O—out, W—walk, S—sacrifice, P—hit by pitch. The program should output for each player the input data followed by the batting average. (Hint: Each batting record is followed by a newline character.)”

You will find information about command line parameters on pp. 682-685 of the text. From the command line parameters, you will only need to use argv[1]. You will use argv[1] as the name of the file in your fopen() function call.

```
[ssdavis@lect1 p3]$ cat baseball1.txt
12 HOOOWSHHOOHPWWHO
4 OSOHHWWOHOHOOO
7 WPOHOHOHOHOHO

[ssdavis@lect1 p4]$ baseball.out baseball1.txt
Player 12's record: HOOOWSHHOOHPWWHO
Player 12's batting average: 0.455

Player 4's record: OSOHHWWOHOHOOO
Player 4's batting average: 0.417

Player 7's record: WPOHOHOHOHOHO
Player 7's batting average: 0.364

[ssdavis@lect1 p3]$ cat baseball2.txt
24 HOOOHWWOHOHPWWHHOOHOWHOOOH
1 OOWOOOOOOH
4 OOWWWFPIOOOOPW
25 HHWWHHWWOOWOHOHO
44 HOOOHWWOHOHOHPWO

[ssdavis@lect1 p4]$ baseball.out baseball2.txt
Player 24's record: HOOOHWWOHOHPWWHHOOHOWHOOOH
Player 24's batting average: 0.500

Player 1's record: OOWOOOOOOH
Player 1's batting average: 0.111

Player 4's record: OOWWWFPIOOOOPW
Player 4's batting average: 0.000

Player 25's record: HHWWHHWWOOWOHOHO
Player 25's batting average: 0.538

Player 44's record: HOOOHWWOHOHOHPWO
Player 44's batting average: 0.286

[ssdavis@lect1 p3]$
```