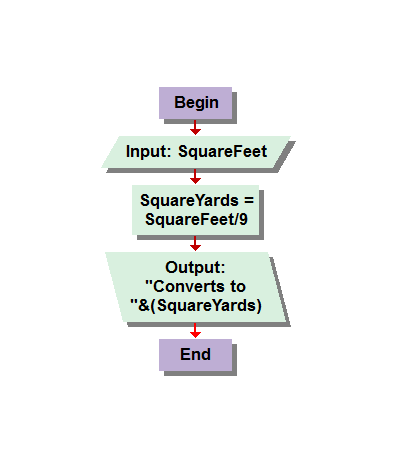
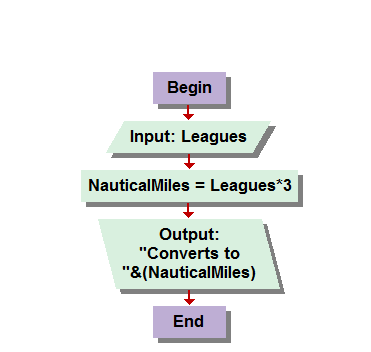
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dimensions in FT | Dimensions in yards (divided by 3) | Square feet  Input | assignemnt (square feet/ square yards) | Square yards |
| 6x6 | 2x2 | 36sq feet | /9 | 4sq yards |
| 12x12 | 4x4 | 144 | /9 | 16 |
| 9x15 | 3x5 | 135 | /9 | 15 |



Problem 1.1. A rose by any other name. This questioned asked to create a function in which you could convert a garden from square feet to square yards… I found the solution to be square feet divided by nine. I found this out by first finding the square feet of a 6x6 garden, 36 square yards, converting the perimeter measurements by to yards, 2x2 , which would be 4 square yards, I then divided 36 by 4 and got nine. So I deduced to convert you would divide by nine.



)Input (Leages) Assignment( x)3 Output( Nautical miles

14 42 42

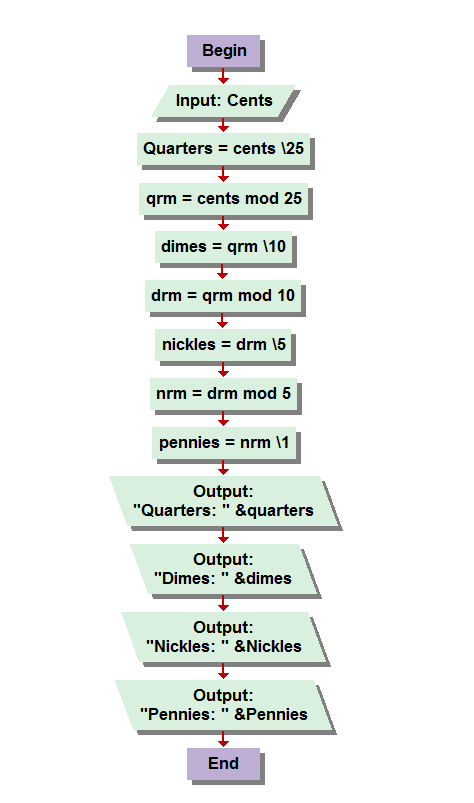
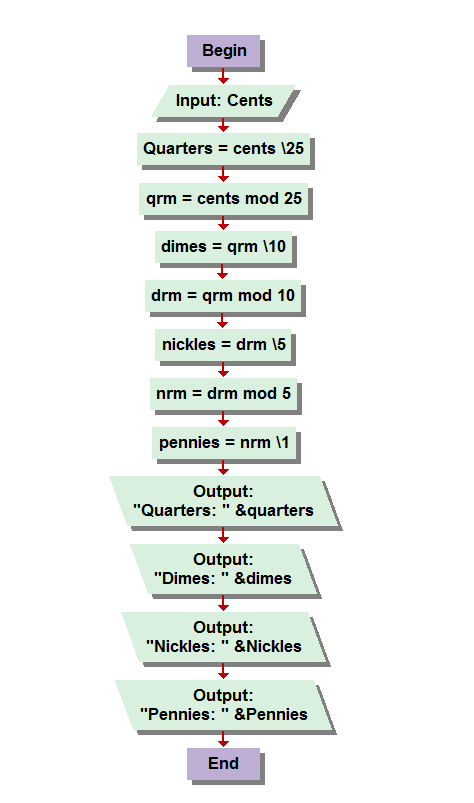
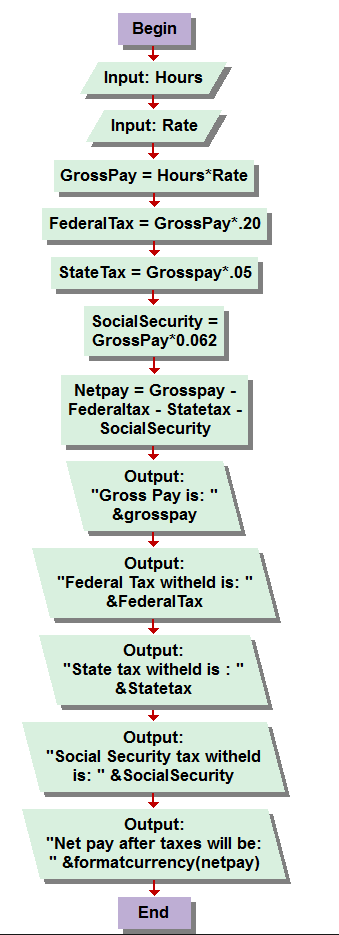
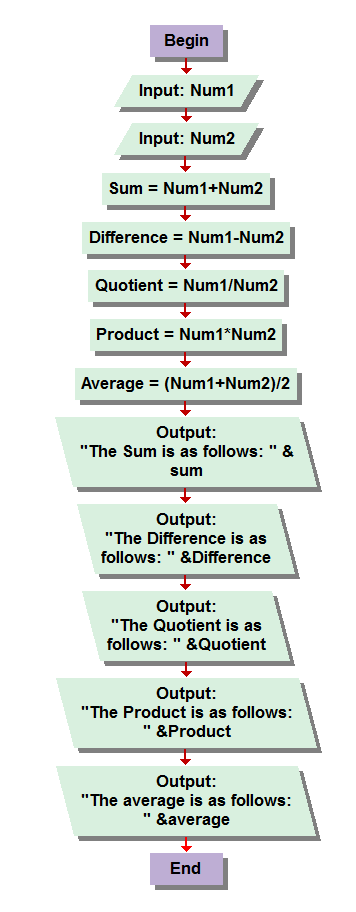
27 81 81

20000 60000 60000

Problem 1.2: Twenty thousand leagues under the sea. I did as the book advised and found the conversion from leagues to nautical miles to be x3. Simply put you would insert the number of leagues and the assignment would produce its product times three. Then the output statement simply states that it, “Converts to” &(nauticalmiles).

|  |
| --- |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| #’s | + | - | / | \* | average |
| 1,2 | 3 | -1 | .5 | 2 | 1.5 |
| 20,10 | 30 | 10 | 2 | 200 | 15 |
| 549, 868 | 1417 | -319 | .632 | 476532 | 705.5 |



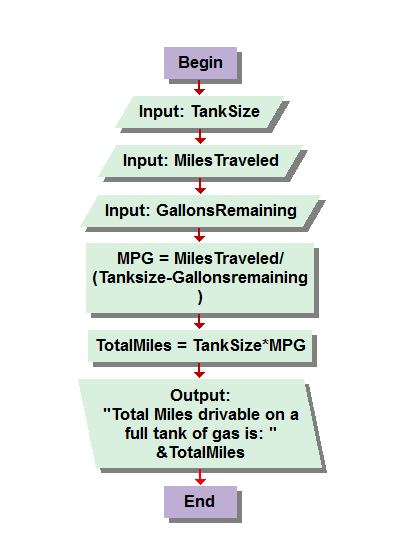
Problem 1.2 Run the numbers. After programming the first two I understood how precise the the output statements had to be, and found further proof with this problem. For some reason I thought I could list all the assignment results in a single output box, which I haven’t confirmed possible either way, but after watching the lecture and heard something about “more than one output statements” I realized how much easier it was going to be to complete this program. I also started this program late at night after work and messed up the programing severely until this morning when I approached it wide awake and alert. I felt very successful and extremely confident when I finished this program and almost empowered to finish the rest, which came quite easily after this one. I know I’m supposed to practice these steps of homework documentation but as you said it does feel rather silly when all the assignments in this exercise are so self-explanatory. I simply created to inputs as the book said, Num1 and Num2 and then added them together, subtracted them, multiplied and divided them, and then added the two together and divided by two for the average. Then created a series of output statements to display the result of each assignment separately in contrast to displaying them all in a list, as I thought was appropriate but could program correctly, it was just giving the me them all in one large number, with in puts of 10 and 5 coming out to 15000 each time.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Hrs | Rate | Gross Pay | Fed.Tax | State  Tax | S.S | Net Pay |
| 40 | 10 | 400 | 80 | 20 | 24.8 | 275.20 |
| 30 | 12.5 | 375 | 75 | 18.75 | 23.25 | 258.00 |
| 40 | 27 | 1080 | 216 | 54 | 66.96 | 743.04 |
| 90 | 600 | 54000 | 10800 | 2700 | D3345 | 37152.00 |

Problem 1.4 Gross pay and net pay: This was a question trying to determine what a pay check would look like after you calculate hours and rate, then deduct taxes withheld. It is a very good concept and could be found very helpful for budgeting your money as you make it, although with myself having 2 kids and a total of 8 federal exemptions, I don’t pay anywhere near this much tax… so my personal paycheck calculator would require some tweaking. Very simple but structured program design, you input your hours, and then your rate and it does the rest. First displaying your gross pay, then your federal tax, your state tax, your social security tax, and then your net pay, which is all of your deductions added up and subtracted from the gross pay. My first attempt to program this I again used parenthesis and it told me something ridiculous to the extent that working forty hours a week at 10 dollars an hour would give me a two thousand plus dollar pay check cause it multiplied things that shouldn’t. Still have to determine exactly when to use and not use parenthesis, I feel like the book lacks a little bit of information on how to exactly word your output statements but I’m learning.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cents | Quarter= Cents\25 | QRM= Quarters Mod 26 | Dimes= Qrm\10 | DRM= Dimes mod  10 | Nickles = Drm \ 5 | NRM= Nickles mod 5 | Pennies= Nrm /1 |
| 86 | 3 | 11 | 1 | 1 | 0 | 1 | 1 |
| 40 | 1 | 15 | 1 | 5 | 1 | 0 | 0 |
| 66 | 2 | 16 | 1 | 6 | 1 | 1 | 1 |

Problem 1.6 Correct Change: Another very neat program, the function is to find the lowest denomination of coinage to equal the correct number of cents for any amount of change needed for, I would imagine, a cashier transaction. This one the solution was hinted at on page eleven, and with the help of the chart you kinda just string an algebraic loop. First Input cents, then you determine quarters by using backslash integer division, \25. Next, u determine the remainder, in this case QuarterReMainder, (qrm) with the cents mod 25, then you repeat the process with dimes= qrm\ 10 and so on and so forth down to pennies and then remainder of nickles being divided anyway by 1 to simply display the value of nickles remaining. Looking back I guess I my last assignment in this program is essentially unnecessary. Or maybe not , regardless it works so I’m not changing it lol. I thought this one could be very useful, although from years of bartending I have extremely accurate and quick mental math abilities when it comes to money in either paper or coin, but the structure of this program has endless possibilities, I would imagine.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tank size | Miles Traveled | Gallons remaing | MPG | Total Miles drivable on a full tank |
| 8 | 60 | 6 | 30 | 240 |
| 12 | 200 | 9 | 66.66 | 800 |
| 24 | 150 | 6 | 11.11 | 200 |

Problem 1.7 Jake’s Problem. I read a lot of confusion about this problem on the message boards but was unable to really feel helpful to anyone unfortunately. I thought this was a very straight forward function. Input for the tank size, miles traveled and gallons remaining. I did the algebra with the figures given, of 6 60 and 8. “  
8 – 6 =2. 2\*X= 60 , X= 30 . 30 x 8 = 240… sometimes I figure things out on paper that I can’t trace correctly in my head, lol. If you saw my scratch paper you would understand. ANYWAY, Input for the three variables, then you take the total number of miles traveled, Divided by, IN PARENTHESE (finally!) the tank size minus the gallons remaining to determine the MPG, miles per gallon. You the multiply this figure by the tank size to get the total number of possible miles driven on a full tank of gas, which is what the question, at least to me, appeared to be asking , though some people seemed to have difficulties.

Okay so please let me know if this homework assignment was done accurately, and I don’t mean in regards to to the programing but more in mention to the layout and formatting of the display of information, charts, ect, just so I know im documenting my progress correctly. I also have questions about out put statements, and how to make them more elaborate. With regards to the simpler conversion functions, I wish I could have figured out a way to make the output more word friendly in the sense that it would display the full conversion. Let me give you an example… I tried origianly for my out put statement in the first problem to say something like, Output: &(squarefeet) “converts to “ &(squareyards) “square yards at a rate of 39to 1” or something like but was unsuccessful . I didn’t read ahead yet so maybe more will be explained in detail about creating more elaborate out put statements when we dive deeper into algorithms but I thoroughly enjoyed these assignements and am quite excited to continue programming!

\_Adam Curcie.