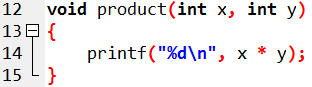
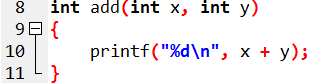
**Problem 16-1**

1. Write down following function definitions (with empty body for now) in your program:
2. Input: nothing, Output: nothing
3. Input: an integer, Ouput: nothing
4. Input: 3 integers, Ouput: nothing
5. Input: 3 integers, Ouput: integer
6. Input: an integer and a float, Ouput: float
7. Write a function which receives 2 Input characters, and returns the larger character in alphabetical order.
8. Consider the following function.

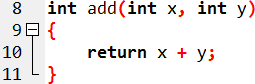


Call the above function in main with following

1. For multiplying two integer variables
2. For multiplying two integer constants 5, and 8
3. For multiplying the integer 7 in loop with values 1, 2, 3, …, 10
4. Try this statement in main: **int result = product(10, 9);** Does your program run? If yes, then what is the output and why? If no, then why?
5. Consider the following function.



1. Try this statement in main: **int result = add(10, 9);** Check this for **output/error.**
2. Move the definition of above function after main and repeat task a above. If there is an error, then also fix the error.
3. Consider the following function.



1. Try this statement in main: **printf(“%d”, add(10, 9));** Check this for **output/error.**
2. Try this statement in main: **printf(“%d”, add(add(10, 9), 9));** Check this for **output/error.**
3. Try this statement in main:

**printf(“%d”, add(add(10, 9), pow(3, 2)));**

Check this for **output/error.**

**Problem 16-2**

**Program Name: Conversion from Celsius to Fahrenheit using functions**

**Purpose:** Usage of functions (and their arguments)

**Problem Statement:** Implement a function “Celsius”, which returns the Celsius equivalent of a Fahrenheit temperature (Hint: The formula for converting from Fahrenheit to Celsius is C = 5/9 \* (F-32);). Use this function to write a program that prints charts showing the Celsius equivalent of all Fahrenheit temperatures between 32 and 212 degrees. Print the output neatly in a table. (Hint: use the loop for counting from 32 to 212, where the counter variable will be the input to the function “Celsius”).

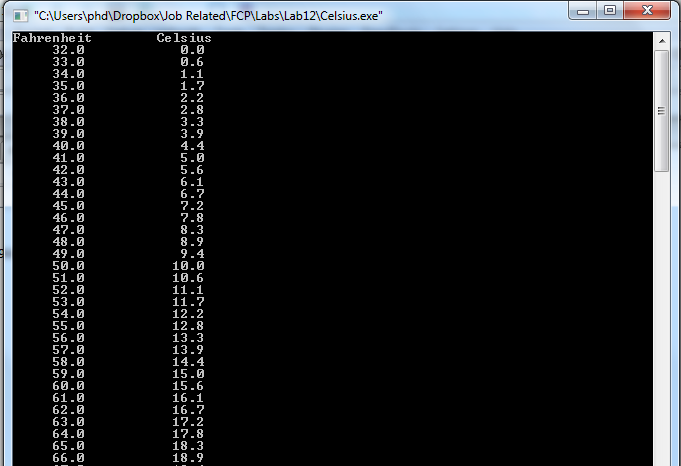
**Pseudocode (for the celsius function)**

1. Input Fahrenheit temperature F
2. Compute 5/9 \* (F-32)
3. Store in Celsius C
4. Return C

**Pseudocode (inside main):**

1. Set temperature F = 32
2. Call Celsius(F) to get corresponding C
3. Display F and C
4. Increase F by 1
5. If F has not reached 212, then go to step 2

**Sample output:**

****

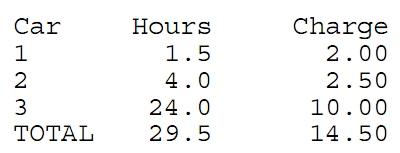
**Problem 16-3**

**Program Name:**

**Purpose:** Usage of functions (and their arguments)

**Problem Statement:** A car park station charges a Rs. 2.00 minimum fee to park for up to 3 hours, and an additional Rs. 0.50 for each hour in excess of three hours. The maximum charge for any given 24-hour period is Rs. 10.00. Assume that no car parks for more than 24 hours at a time.

Write a C program that will calculate and print the parking charges for each of 3 customers who parked their car in the car park yesterday. The program should accept as input the number of hours that each customer was parked, and output the results in a neat tabular form, along with the total receipts from the three customers:



The program should use the function calculate\_charges to determine the charge for each customer. (Hint: First, identify the input and output of the function? Then, implement the function using above described rules and call this function for each customer.)

**Pseudocode (for calculate\_charges function)**

**Input:** number of hours H

**Ouput:** total charges C

If H <= 3

Set C to H\*2

Else

Set excess charges E = (H-3) \* 0.50

Set C to 3\*2

C = C + E.

End if

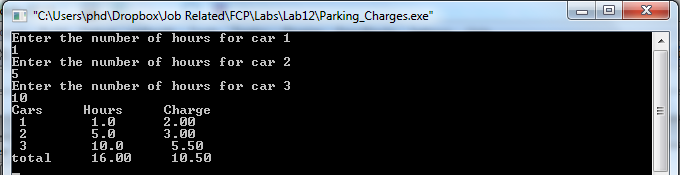
If C > 10.00

Set C = 10.00

End if

Return C

**Sample output:**

****

**Lab 17.1:** A 2D array of size MxN is given. The purpose of this program is to shift a row up or down in the 2D array. For example, if the input array is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 5 | 2 | 9 | -2 |
| -5 | 10 | 5 | 66 | 1 |
| 0 | 0 | 56 | 6 | 7 |
| 89 | -15 | -11 | 0 | 25 |

Then after shifting the 3rd row UP, new 2D array should be:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 5 | 2 | 9 | -2 |
| 0 | 0 | 56 | 6 | 7 |
| -5 | 10 | 5 | 66 | 1 |
| 89 | -15 | -11 | 0 | 25 |

Write a program, in which user enters the row number, which should be shifted up. The program should perform the shift operation using following algorithm. (**Hint:** you don’t need to run a nested loop, a single loop would be enough.)

**Algorithmic hint:** If we have to shift up row i, then swap each individual element of ith row with (i-1)th row.

**Home Task 1:** Modify the above program so that a row can be shifted up multiple times. The program asks the user how many times he wants to shift a particular row. E.g. if the last row (row number 3) is shifted 3 times, then the above array becomes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 89 | -15 | -11 | 0 | 25 |
| 1 | 5 | 2 | 9 | -2 |
| -5 | 10 | 5 | 66 | 1 |
| 0 | 0 | 56 | 6 | 7 |

(Hint: here you will have to use nested loop)

**Home Task 2:** Write a program that simulates coin tossing. For each toss of the coin the program should print Heads or Tails. Let the program toss the coin 100 times, and count the number of times each side of the coin appears. Print the results. The program should call a separate function flip that takes no arguments and returns 0 for tails and 1 for heads. [*Note:* If the program realistically simulates the coin tossing, then each side of the coin should appear approximately half the time for a total of approximately 50 heads and 50 tails.] (Note: the general formula for generating a random number between a and b with interval = x is **a + ( rand() % ((b+1-a)/x) ) \*x**

*Lab Task 17.1: A player rolls two dice. Each die has six faces. These faces contain 1, 2, 3, 4, 5, and 6 spots. After the dice have come to rest, the sum of the spots on the two upward faces is calculated. If the sum is 7 or 11 on the first throw, the player wins. If the sum is 2, 3, or 12 on the first throw (called “craps”), the player loses (i.e., the “house” wins). If the sum is 4, 5, 6, 8, 9, or 10 on the first throw, then that sum becomes the player’s “point.” To win, you must continue rolling the dice until you “make your point.” The player loses by rolling a 7 before making the point.*

**Problem 17-2**

You have to write a program using functions, which can print a rectangle of stars (Asterisks) of given width and height. The function should receive the width and height of the rectangle and print the asterisks rectangle accordingly. Test this function by calling in main with different width and height values. (HINT: you will have to use nested loops to print the asterisks rectangle)