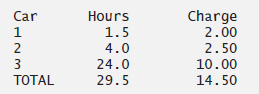
# Quizes

1. (World Population Growth) World population has grown considerably over the centuries.  
   Continued growth could eventually challenge the limits of breathable air, drinkable water, arable  
   cropland and other limited resources. There is evidence that growth has been slowing in recent years and that world population could peak some time this century, then start to decline.  
   For this exercise, research world population growth issues online. Be sure to investigate various  
   viewpoints. Get estimates for the current world population and its growth rate (the percentage by  
   which it is likely to increase this year). Write a program that calculates world population growth  
   each year for the next 75 years, using the simplifying assumption that the current growth rate will stay constant. Print the results in a table. The first column should display the year from year 1 to year 75. The second column should display the anticipated world population at the end of that year.  
   The third column should display the numerical increase in the world population that would occur  
   that year. Using your results, determine the year in which the population would be double what it  
   is today, if this year’s growth rate were to persist.
2. (Parking Charges) A parking garage charges a $2.00 minimum fee to park for up to three  
   hours. The garage charges an additional $0.50 per hour for each hour or part thereof in excess of three hours. The maximum charge for any given 24-hour period is $10.00. Assume that no car parks for longer than 24 hours at a time. Write a program that calculates and prints the parking charges for each of three customers who parked their cars in this garage yesterday. You should enter the hours parked for each customer. Your program should print the results in a neat tabular format and should calculate and print the total of yesterday’s receipts. The program should use the function calculateCharges to determine the charge for each customer. Your outputs should appear in the following format:



1. (Calculating Number of Seconds) Write a function that takes the time as three integer arguments (hours, minutes and seconds) and returns the number of seconds since the last time the clock “struck 12.” Use this function to calculate the amount of time in seconds between two times, both of which are within one 12-hour cycle of the clock.
2. (Coin Tossing) 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.]
3. Write a recursive function f with one positive int parameter n. The function prints out

2n-1 integers as follows:

f(1) outputs 1

f(2) outputs 1 2 3

f(3) outputs 1 2 3 4 5 6 7

f(4) outputs 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

1. Write a recursive function void **listNumbers**(int start, int end) that outputs the numbers from start to end to console. Write one version that outputs the numbers in ascending order, and another that outputs them in descending order.
2. Write a recursive function int **min**(int[] a, int start, int end) that returns the smallest element between the indices start and end in the parameter array a.
3. Write a recursive function **Binfun**() prints binary equivalent of n.   
   For example, if n is 21 then prints 10101.
4. Write a recursive function that calculate Ab where Ab=Ab-1\*A, Then use this function to calculate an array R which is equal to A to power b where R and A and b are on dimensional arrays.

Example : I/P A= {1, 2, 3, 4}, b={2,3,4,5} the output is R={1, 8, 81, 1024}

1. Write a function **SameRow** to return the indices of repeated rows of a 2D array. Return the indices in a 1D array
2. Write function **Transpose** to switch the content of rows and columns of a 2D array
3. Write a function **Remove** that removes the values of specific range,

specified by starting index and last index, in an array.

Example Remove({9,3,5,4,7,5},1,3)

The sent array {9,3,5,4,7,5} should be modified into {9,7,5,0,0,0}

1. Write a recursive function **gcd** that returns the greatest common divisor of x and y
2. Write a function **getWords** that takes a pointer to a null terminated array of characters (s1) and will return the list of words contained in that string and count words.
3. Write a function **getRepeatedWords** that takes a pointer to a null terminated array of characters (s1) and will return repeated words contained in that string, count repeated words and replace repeated words with (#).