

HOMEWORK 2

1. Write a function named `middle` that finds the middle value of three numbers. The function takes three numbers as parameters and returns the middle value. The middle value means the value that is greater than the smallest value and less than the largest value.
2. Write a function named `factors` that finds the factors of a number. The function takes a number as a parameter and returns a string. The string contains the factors separated by commas and spaces. The factors include 1 and the number. The number is a positive integer.
3. Write a function named `tax` that computes the tax according to the following tax rules. The tax depends on the income and the marital status. If single and income is less than 30000 then tax rate is 20%. If single and income is greater or equal to 30000 then tax rate is 25%. If married and income is less than 50000 then tax rate is 10%. If married and income is greater or equal to 50000 then tax rate is 15%. The function takes the income and the marital status as parameters and returns the tax amount. The income is a positive number. The status is a string `single` or `married` in lower or upper case.
4. Write a function named `stdDev` that computes the standard deviation of a set of numbers. The function takes the numbers as a variable number of parameters and returns the standard deviation. The standard deviation of n numbers x_1, x_2, \dots, x_n is the square root of $[(x_1 - m)^2 + (x_2 - m)^2 + \dots + (x_n - m)^2] / n$ where m is the average of the numbers. If there are less than two numbers the standard deviation is 0.
5. Write a function named `compoundInterest` that computes the compound interest. If p amount is invested for n years with interest rate r and the money is compounded annually then the final amount will be $p(1 + r/100)^n$. The function takes initial amount p , interest rate r which is between 0 and 100, and the number of years n as parameters and returns the final amount. The parameter values are all positive.
6. Write a function that takes a character as a parameter and returns its type. The possible types are `upper`, `lower`, `digit`, and `other`. Upper means an upper case letter. Lower means a lower case letter. Digit means a numerical digit. Other means all other characters. The return value is a string in lower case. Regular expressions cannot be used.
7. Write a function named `createIdPassword` that takes a last name and a first name as parameters and returns an object containing an id and a password. The id is the first letter of the first name followed by the last name. The password is the first letter of the first name followed by the last letter of the first name followed by the first three letters of the last name followed by the length of the first name followed by the length of the last name. Both id and password are all upper case. For example if the first name is `John` and the last name is `Maxwell` then the id is `JMAXWELL` and the password is `JNMAX47`. The returned object has two properties namely `id` and `password`, and their values are set to the id and password that are created.

8. Write a function named `removeDuplicates` that takes an array of strings as parameter and returns a duplicate free array. The calling/parameter array does not change. The unique strings of the calling array are placed in a newly created array and it is returned. For example if the calling array is `[tree, cat, box, cat, dog, tree, tree]` then the returned array is `[tree, cat, box, dog]`.

9. Write a function named `mySort` that takes three arrays and sort/rearrange them in parallel. The three arrays contain information about students. The first array contains last names. The second array contains gpa's. The third array contains zip codes. The sorting is performed in the ascending order of last names. The function changes the calling/parameter arrays. Built in sorting methods cannot be used. Write custom code using selection sort algorithm.

10. Write a function named `myReverse` that takes a line of words as a parameter, reverses the order of words, reverses the order of characters in every other word, and returns the result. The input and the output are strings. The words are separated by single spaces in the input and the output. For example if the line of words is `tree is big green` then the result is `neerg big si tree`.

11. Write a function named `ApplyFunctionToArray` that takes a function `f` and an array `p` as parameters. The `f` is a function that takes a number as a parameter and returns a number as output. The `p` is an array of numbers. The function calls `f` on each element of `p` and replaces the element with the output of `f`. For example if `f` is a square function then all the values in `p` will be squared. The function changes the calling/parameter array.

12. Write a class called `Student`. The class has two properties called `name` and `gpa`. The class has a constructor that takes a name and a gpa and set them to the properties of the class. The class has `getName`, `getGpa`, `setName`, and `setGpa` methods that get and set name and gpa. The class has `isHonors` method which returns `true/false` depending on whether gpa is above or below 3. The class has `toString` method that returns a string containing name and gpa.

13. Write a function named `university` that takes a string as a parameter and decides whether it is a valid university id. The university id format is `E-0xxy-9yyx`. Write a method named `phone` that takes a string as a parameter and decides whether it is a valid phone number with area code 313 or 248 or 734. The phone number format is `xxx-xxx-xxxx`. Here `x` is a digit and `y` is a lower case letter. These functions return `true/false`. The code must be based on regular expressions.

14. Write a function named `fullName` that takes a string as a parameter and decides whether it is a full name of a person. The full name format is `Prefix First M. Last`. Prefix is Mr, Mrs, or Ms. First is the first name that begins in upper case letter followed by one or more lower case letters with a total length of at least two. M is the middle initial which is a single upper case letter. Last is the last name which has the same format as the first name. There is a dot after the middle initial. There is one space between prefix, first name, middle initial, and last name. The function returns `true/false`. The code must be based on regular expressions.

The functions in questions 1-14 are in one file named program.js. This file should only have the required functions and nothing else.

15. Create a web page that calculates the compound interest. The inputs to the calculation are the initial amount, interest rate, and number of years. The output is the final amount. If p amount is invested for n years with interest rate r then the final amount will be $p(1 + r/100)^n$. There are three text boxes for the inputs. These text boxes have appropriate labels on the left. The text boxes and the labels are properly aligned. There is a button below the boxes/labels to calculate the final amount. The final amount is displayed below the button. The page has an appropriate title. Everything is centered on the page. Use appropriate css style.

16. Create a web page that calculates the cost of attending a university. The inputs to the calculation are the number of credits, undergraduate/graduate status, in state/out of state status, and selection of dorm/dining/parking options. The number of credits is entered in a text box. The undergraduate/graduate status is chosen from radio buttons. The in state/out of state status is also chosen from radio buttons. The dorm/dining/parking options are selected from check boxes. There are appropriate labels. There is a button below to calculate the total cost. The total cost is displayed below the button. The page has an appropriate title. Everything is centered on the page. Use appropriate css style. The in state tuition is \$200 for undergraduates and \$300 for graduates. The out of state tuition is twice the in state tuition. The dorm, dining, and parking have fixed costs of \$1000, \$500, and \$200 respectively.

17. Create a web page that makes id and password for a user. The inputs are last name, first name, and phone number. The id is the first letter of the first name followed by the last name. The password is the last letter of the first name followed by the first three letters of the last name followed by the last four digits of the phone number. Both id and password are all upper case. For example if the first name is John, last name is Maxwell, and phone number is 248-320-5195 then the id is JMAXWELL and the password is NMAX5195. There are three text boxes for the inputs. The boxes have place holders instead of labels. Limit the number of characters in each name to 10. There is a button below the boxes to calculate the id and password. The id and password are displayed below the button. The three inputs are checked for errors when the button is clicked. The last and first names are strings of lower or upper case letters of length at least two. The phone number format is xxx-xxx-xxxx. If an input has an error, an error message is shown on the right in red color. The error message includes the correct input format. The error message does not appear if there is no error. If there is an error then the id and password are not displayed. The page has an appropriate title. Everything is centered on the page. Use appropriate css style.

18. Repeat the id/password question with the following changes. When the mouse moves over an input box, a dynamic message indicating the format of that input appears on the right. When the mouse moves out of a box, its dynamic message disappears. When a box becomes focused its background becomes white. There is no submit button to calculate the id and password. Whenever a change event occurs in a box, the inputs from all three boxes are read and checked for errors. If there is an error then the background of the corresponding box changes to red, and the id and password are not displayed. If there are no errors then the id and password are

displayed. When an error is corrected in a box its background becomes white. When the id and password are displayed, the backgrounds of all boxes are white and there are no messages.

19. This is related to the laptop project. Add the following features to the page. When the mouse moves over a paragraph, the paragraph's background color and font family change. When the mouse moves over a heading, its font size changes. When the mouse over an image, its border changes. When the mouse moves out of a paragraph, heading, or image they go back to their original states. Choose appropriate css style changes. The style changes must be done by javascript code. Decide how to write event handlers and what parameters to use. Use normal images without opacity changes.

20. This is related to the resume project. Add the following feature to the resume. When the resume is clicked anywhere, a box appears at the clicked location. The box contains a link to a company. When the resume is clicked again anywhere, the box disappears. When the resume is clicked again anywhere, the box appears at the clicked location containing a link to another company. This continues. In other words, the box alternately appears and disappears with clicks. The company of a click is randomly chosen from ten companies you like to work. Use appropriate css style for box and link.

21. This is related to the EMU table project. Add the following feature to the page. A small box containing the word EMU always follows the mouse pointer. The box touches the pointer and it is located below the pointer so that it does not hide anything above the pointer. Initially the box is not on the page. As soon as the mouse pointer enters the page, the box starts to follow the pointer. The box follows the pointer regardless of the pointer moves or not. When the mouse pointer exits the page, the box disappears. Use appropriate css style for box and word.

22. This is related to the personalized web portal project. Add the following features to the portal. When the page starts/loads, the quote near the left margin is randomly chosen. The quote is not fixed. The quote is randomly chosen from ten favorite quotes of yours. The current time is shown near upper right corner. The time shows hours, minutes, and seconds separated by colons. The time is updated every ten seconds using an animation loop. Use appropriate css style for quote and time.

23. Create a web page to play a simple game. The game is played in a large square area. There are two players. One player is a small green square. The other player is a small red square. Both players have the same size. The green player is randomly jumping from place to place within the playing area. The green player can jump from any point to any point. The movement of the green player is controlled by an animation loop. The red player is trying to catch the green player. The red player can only jump in left, right, up, or down directions by a fixed distance. There are five buttons located horizontally below the playing area. The user clicks on the start button to start the game. The user clicks on left, right, up, or down button to move the red player in left, right, up, or down direction. When a direction button is clicked the red player jumps in that direction by the fixed distance. The game continues even if the red player catches the green player. You need to choose the following: size of players, how often the green player changes its position, jumping distance of the red player. You need to carefully choose these values so that the game does not become too difficult or too easy. Use appropriate css style for the game.

Only those features discussed in the class/lecture notes may be used in the homework. Test the programs on the Chrome browser.