1. Векторизуйте наступне (перепишіть код ефективно):

*i=0;*

 *for inc=0: 0.5: 3*

 *i=i+1;*

*myvec(i)=sqrt(inc);*

*end*

1. Векторизуйте наступне (перепишіть код ефективно):

*[r c]=size(mat);*

 *newmat=zeros(r,c);*

 *for i=1:r*

*for j=1:c*

*newmat(i,j)=sign(mat(i,j));*

*end*

*end*

1. Векторизуйте наступний код. Припустимо, що v є векторною змінною, яка була ініціалізована.

*for i=1:length(v)*

 *newv(i)=sqrt(v(i));*

*end*

*newv % Just for display*

1. Перепишіть це як один оператор, який виконає те саме для матричної змінної mat (векторизуйте цей код):

*[r c]=size(mat);*

*for i=1:r*

*for j=1:c*

*mat(i,j)=mat(i,j)\*2;*

*end*

*end*

1. Напишіть програму гру «Вгадай число». Програма генерує випадкове ціле число в заданому діапазоні, і користувач (гравець) повинен вгадати число. Програма дозволяє користувачам грати скільки завгодно разів; наприкінці кожної гри програма запитує, чи бажає гравець грати знову. Основний алгоритм такий:

1. Програма запускається та друкує інструкції на екрані.

2. Для кожної гри:

– програма генерує нове випадкове ціле число в діапазоні від MIN до MAX. Розглядайте MIN і MAX як константи; почніть з ініціалізації їх, наприклад, 1 і 100.

* існує цикл, який пропонує гравцеві вгадувати, доки гравець правильно не вгадає ціле число.
* для кожного припущення користувача програма друкує на екран, чи було припущення гравця занадто менше, занадто вище або правильним.

3. В кінці (коли вгадано ціле число):

* вивести загальну кількість відгадок для цієї гри
* надрукувати повідомлення про те, наскільки добре гравець виконав цю гру (наприклад, гравцеві знадобилося занадто багато часу, щоб вгадати число, чи гравець був чудовим тощо). Для цього вам доведеться визначитися з діапазонами для своїх повідомлень і обґрунтувати своє рішення в коментарях у програмі.
1. Після того, як усі ігри зіграно, роздрукуйте підсумок із зазначенням середньої кількості відгадок.
2. Створіть символьну векторну змінну таким чином: word='AALMTB';

Використовуйте функцію randperm для випадкового переміщування символів і збереження результату в змінній «newword», наприклад результат може бути таким:

newword= 'TALMBA'

Тепер помістіть це в цикл. 20 разів, випадковим чином перемішайте символи та відобразіть результати. Чи отримали Ви комбінацію «MATLAB»?

Скільки разів така комбінація букв створиться при 100 повтореннях циклу?

1. Побудуйте наступний графік двома способами:
* з використанням циклу for-end та за потреби операторів умовних переходів.
* без використання циклів та операторів умовних переходів.

Крок по осі x – 0,01



1. Створіть вектор-строку зі значеннями функції sin() від 0 до 2\*pi з кроком 0.001

Створіть три вектори-строки випадкових чисел, розміром як створений sin(), розподілених за нормальним законом – normrnd() – з нульовим математичним сподіванням та з різними СКВ (середній квадрат відхилення випадкової величини): 0.1, 0.25, 0.8. Не повний формат виклику такий: noise = normrnd(0, параметр СКВ, розмірність вектора)

1. Використайте цикл for для побудови графіків шуму з використанням функції subplot(). Тобто після завершення циклу буде побудовано один графік з трьома областями.
2. Додайте до функції sin() згенерований шум
3. Побудуйте ще графіки з використанням циклу for та функції subplot(), але тепер для суми шуму та синуса.
4. Напишіть функцію, яка отримує матрицю як вхідний аргумент і друкує випадковий стовпець із матриці.
5. Напишіть програму для обчислення положення снаряда в заданий момент часу t. Для початкової швидкості v0 та кута вильоту θ0 положення визначається координатами x і y таким чином (примітка: прискорення вільного падіння g становить
9,81 м/с2):



Програма повинна ініціалізувати змінні для початкової швидкості, часу та кута вильоту. Потім слід викликати функцію для пошуку координат x і y і надрукувати результат в командне вікно.

1. Напишіть функцію, яка отримуватиме координати x і y двох точок і повертатиме Евклідову відстань між ними, а також Манхеттенську відстань.