

## PART 1

```
x = [1, 3, 5, 10];
y = addeven(x);
disp(y);
```

13

```
x = 2:2:1000;
y = addeven(x);
disp(y);
```

125500

## PART 2

```
% Explanation of the Program
% L=length(x);: This line calculates the length of the array x and stores it in the
variable L.
% This means L will hold the number of elements in the array x.
% for i=1:L: This line starts a loop that iterates over each index i from 1 to L.
% if x(i) < 0: Inside the loop, this condition checks if the current element x(i)
is less than 0.
% x(i)=-1;: If the condition is true (i.e., the element is negative), this line
sets the current element x(i) to -1.

% Same functionality without using a for loop: x(x < 0) = -1;
% Explanation of the New Code
% x < 0: This creates a logical array where each element is true (1) if the
corresponding element in x is less than 0, and false (0) otherwise.
% x(x < 0) = -1;: This line uses logical indexing to directly set all elements of x
that are negative to -1.
```

## PART 3

```
x = 1:1000;
mean_x = mean(x);
variance = sum((x - mean_x).^2) / length(x);
disp(['Varyans: ', num2str(variance)]);
```

Varyans: 83333.25

## PART 4

```
x = rand(1, 1000);
f = zeros(size(x));
f(x >= 0.2) = 0.2;
f(x < 0.2) = -0.2;
disp(f);
```

```
0.2000    0.2000   -0.2000    0.2000    0.2000   -0.2000    0.2000    0.2000    0.2000    0.2000   -0.2000    0
```

## FUNCTIONS

```
function y = addeven(x)
    even_indices = 2:2:length(x);
    y = sum(x(even_indices));
end
```