1. When the program starts the program asks the user for the following parameters:
a. Sampling frequency of signal.
b. Start and end of time scale
c. Number of the break points and their positions (i.e. the points that the signal definition rule
changes).

```
% inputs from users start and end and FS and BK
tstart = input ('Enter the start time \n')
tend = input ('Enter the end time \n')
fs = input ('Enter sample frequency \n')
Bk = input ('Enter the number of breake points \n')

% define variable t in break point = 0
if Bk == 0
    t=[tstart:(1/fs):tend];  % t is number of samples from start to end

 % define variable t in break point ~= 0 and take new BK points
elseif Bk ~= 0
    p=[1:Bk+2];
    p(1)=tstart;
    p(Bk+2)=tend;
    for Counter = 1:Bk
        p(Counter+1) = input (' please enter the position of the next Break Point\n ');
    end
    t=[p(1):(1/fs):p(2)];
end
```

2. According to the number of break points the program asks the user at each region to enter the specifications of the signal at this region **Which are:**

a. DC signal: Amplitude.
b. Ramp signal: slope – intercept.
c. General order polynomial: Amplitude-power – intercept.
d. Exponential signal: Amplitude – exponent.
e. Sinusoidal signal: Amplitude – frequency – phase.

```
% Type of signal ,if BK = 0 (one function )
Signal = input ('Select Signal typ \n (1) for DC signal \n (2) for Ramp signal \n (3) for general order polynomial signal \n (4) for Exponential signal \n (5) for sinusoidal signal\n')
switch Signal
%   DC signal
case (1)
    amplitude = input ('Please enter the amplitude of the signal\n')
    y=amplitude*ones(1,length(t));

%  Ramp signal
case(2)
    slope = input ('Enter the slope of the ramp signal \n')
    intercept = input ('Enter the intercept of the ramp signal \n')
    y = slope*t+intercept

%     general order polynomial signal
case(3)
    power = input ('Enter the highest power of the signal \n')
    intercept = input ('Enter the intercept of the signal (the free term) \n')
    y=intercept*ones(1,length(t));
    c=[1:power];
%    calculate coefficents
    for r = 1:power
        c(r)= input('Enter coefficient starting from the lowest power \n');
  end

%     calculate function of polynomial
    for q = 1:power
        y2 = c(q)*(t.^q);
        y = y+y2;   % add one  intercept only
    end
%   Exponential signal
case(4)
    amplitude = input ('Please enter the amplitude of the signal \n')
    exponent = input ('Please enter the exponent of the signal \n')
    y= amplitude*exp(exponent*t);
```

```
%  sinusoidal signal
case(5)
    amplitude = input ('Please enter the amplitude of the signal \n')
    frequency = input ('Please enter the frequency of the signal\n')
    phase = input ('Please enter the phase shift of the signal\n')
    y= amplitude*sind(2*pi*frequency*t+phase);
end

% Type of signal for non-zero break points  (morer than one function )
if Bk ~= 0
    for n = 1:Bk
        Signal = input ('Enter signal type \n (1) for DC signal \n (2) for Ramp signal \n (3)
for general order polynomial signal \n (4) for Exponential signal \n (5) for sinusoidal
signal\n')
        switch Signal
            case(1)
                amplitude = input ('Please enter the amplitude of the signal \n')
                t1=[p(n+1):(1/fs):p(n+2)];
                y1=amplitude*ones(1,length(t1));
            case (2)
                slope = input ('Please enter the slope of the ramp signal \n')
                intercept = input ('Please enter the intercept of the ramp signal \n')
                t1=[p(n+1):(1/fs):p(n+2)];
                y1 = slope*t1+intercept
            case(3)
                power = input ('Please enter the highest power of the signal \n')
                intercept = input ('Please enter the intercept of the signal (the free term) \n')
                t1=[p(n+1):(1/fs):p(n+2)];
                y1=intercept*ones(1,length(t));
                c=[1:power];
                for r = 1:power
                    c(r)= input('please enter the next coefficient starting from the lowest power
\n');
                end
                for q = 1:power
                    y2 = c(q)*(t1.^q);
                    y1 = y1+y2;
                end
            case(4)
                amplitude = input ('Please enter the amplitude of the signal \n')
                exponent = input ('Please enter the exponent of the signal \n')
                t1=[p(n+1):(1/fs):p(n+2)];
                y1= amplitude*exp(exponent*t1);
```

```
        case(5)
            amplitude = input ('Please enter the amplitude of the signal  \n')
            frequency = input ('Please enter the frequency of the signal  \n')
            phase = input ('Please enter the phase shift of the signal \n')
            t1=[p(n+1):(1/fs):p(n+2)];
            y1 = amplitude*sind(2*pi*frequency*t1+phase);
    end

  y=[y(1:end-1) y1];
  t=[t(1:end-1) t1];
  end
end
```

3. Display the resulting signal in time domain.

```
% Graph for continous or discrete
  P = input ('Enter the graph form \n (1) for countinous \n (2) for discrete\n');
  if P == 1
     plot(t,y)
     grid on
  elseif P == 2
     stem (t,y)
     grid on
  end
```

4. the program asks the user if he wants to perform any operation on the signal
a. Amplitude Scaling: scale value.
b. Time reversal.
c. Time shift: shift value.
d. Expanding the signal: expanding value
e. Compressing the signal: compressing value
f. None

```
%   operation on signal
 go = true;
  while go
   operation=input('Select an operation \n (1) Amplitude scaling \n (2) Time reversal \n
(3) Time shift \n (4) Expanding the signal \n (5) Compression the signal \n (6) Exit  \n')
    switch operation
        case(1)
            scale = input('enter the scale value\n')
            tn = t ;
            yn = scale*y ;

        case(2)
            tn = -t ;
            yn = y ;

        case(3)
            shift = input('Please enter the shift value\n')
            tn=t+shift;
            yn=y;

        case(4)
            Expanding = input('Please enter the Expanding value (the coefficient of t in your
equation)\n')
            tn=t/Expanding;
            yn=y;

        case(5)
            Compression = input('Please enter the Compression value (the coefficient of t in
your equation)\n')
            tn=t/Compression;
            yn=y;

         case(6)
             go=false;
    end
  end
```

5. Display the new signal in time domain

```
%    Graph to operation signal continous or discrete
P = input ('(1) for countinous \n (2) for discreate\n');
if P == 1
    plot(tn,yn)
    grid on
elseif P == 2
    stem (tn,yn)
    grid on
end
```