MINI PROJECT 1

Team members

1- Seif Mohamed el said 102
2- Mohamed Said Mohamed 179
3- Mahmoud Sameh Mohamed 196
4- Mohamed Tarek Tolba 182

First, we start the program (type main) and answer some question to set up break points.

```
>> main
Welcome to 'The General Signal Generator' project.
Define sampling frequency:  50
Define start time:  0
Define end time:    15
Break points number:    4
Break point #1 time:    3
|| Alligned to sample point at time 3
Break point #2 time:    6
|| Alligned to sample point at time 6
Break point #3 time:    9
|| Alligned to sample point at time 9
Break point #4 time:    12
|| Alligned to sample point at time 12
```

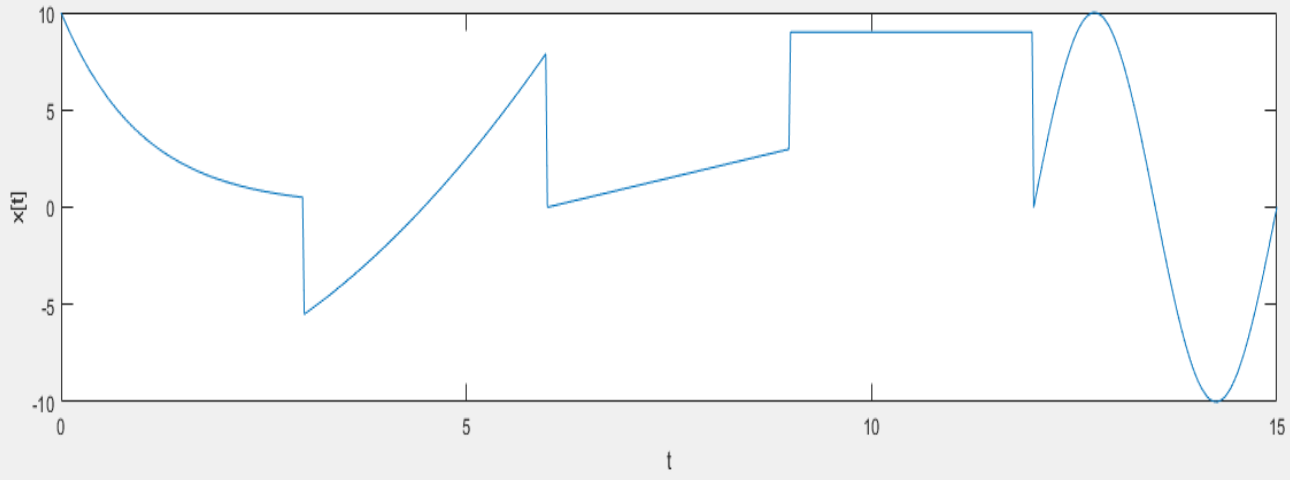
Then we choose the function/s that we intend to get plotted and enter information that we get asked for.

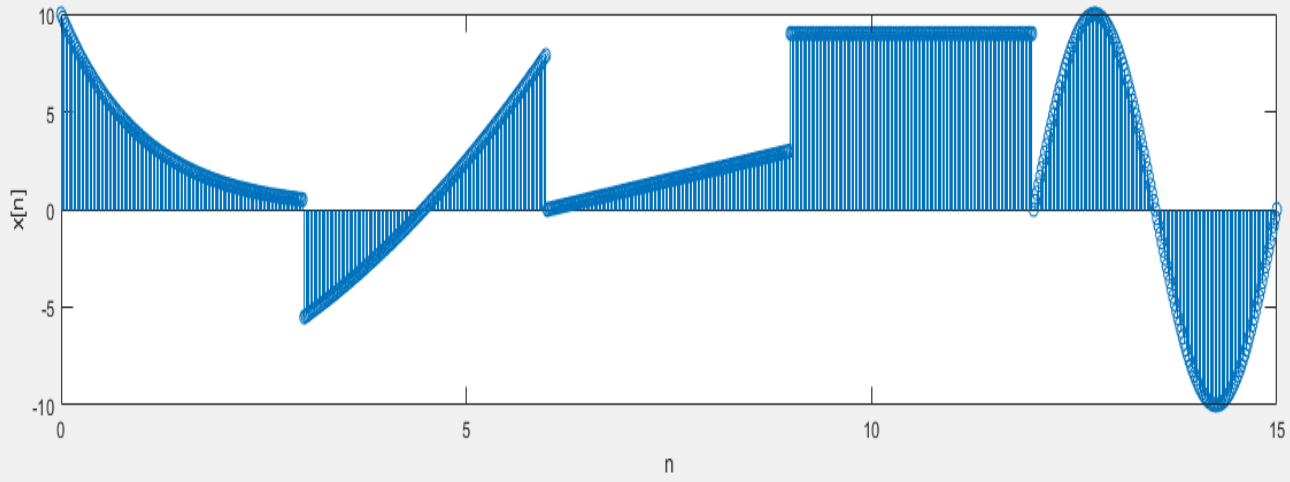
the following code has been set to show all types of functions the program has to offer. The following entered data has been chosen randomly.

```
    ++++ Signal Definition Rules ++++

Choose a definition rule for region #1 , [ 0 , 3 ]
1)    DC Signal.
2)    Ramp Signal.
3)    GOP Signal.
4)    Exponential Signal.
5)    Sinusoidal Signal.
Choose [1 ~ 5]: 4
Exponential Signal Amplitude:    10
Exponential Signal Exponent:    -1

Choose a definition rule for region #2 , [ 3 , 6 ]
1)    DC Signal.
2)    Ramp Signal.
3)    GOP Signal.
4)    Exponential Signal.
5)    Sinusoidal Signal.
Choose [1 ~ 5]: 3
GOP Signal Order:    2
t^2 Amplitude:  0.5
t^1 Amplitude:  0
GOP Signal Intercept:    -10
```

```
Choose a definition rule for region #3 , [ 6 , 9 ]
1)    DC Signal.
2)    Ramp Signal.
3)    GOP Signal.
4)    Exponential Signal.
5)    Sinusoidal Signal.
Choose [1 ~ 5]: 2
Ramp Signal Slope:  1
Ramp Signal Intercept:  -6

Choose a definition rule for region #4 , [ 9 , 12 ]
1)    DC Signal.
2)    Ramp Signal.
3)    GOP Signal.
4)    Exponential Signal.
5)    Sinusoidal Signal.
Choose [1 ~ 5]: 1
DC Signal Amplitude:    9
```

```
Choose a definition rule for region #5 , [ 12 , 15 ]
1)    DC Signal.
2)    Ramp Signal.
3)    GOP Signal.
4)    Exponential Signal.
5)    Sinusoidal Signal.
Choose [1 ~ 5]: 5
Sinusoidal Signal Amplitude:     10
Sinusoidal Signal Frequency:     1/3
Sinusoidal Signal Phase:     0
```



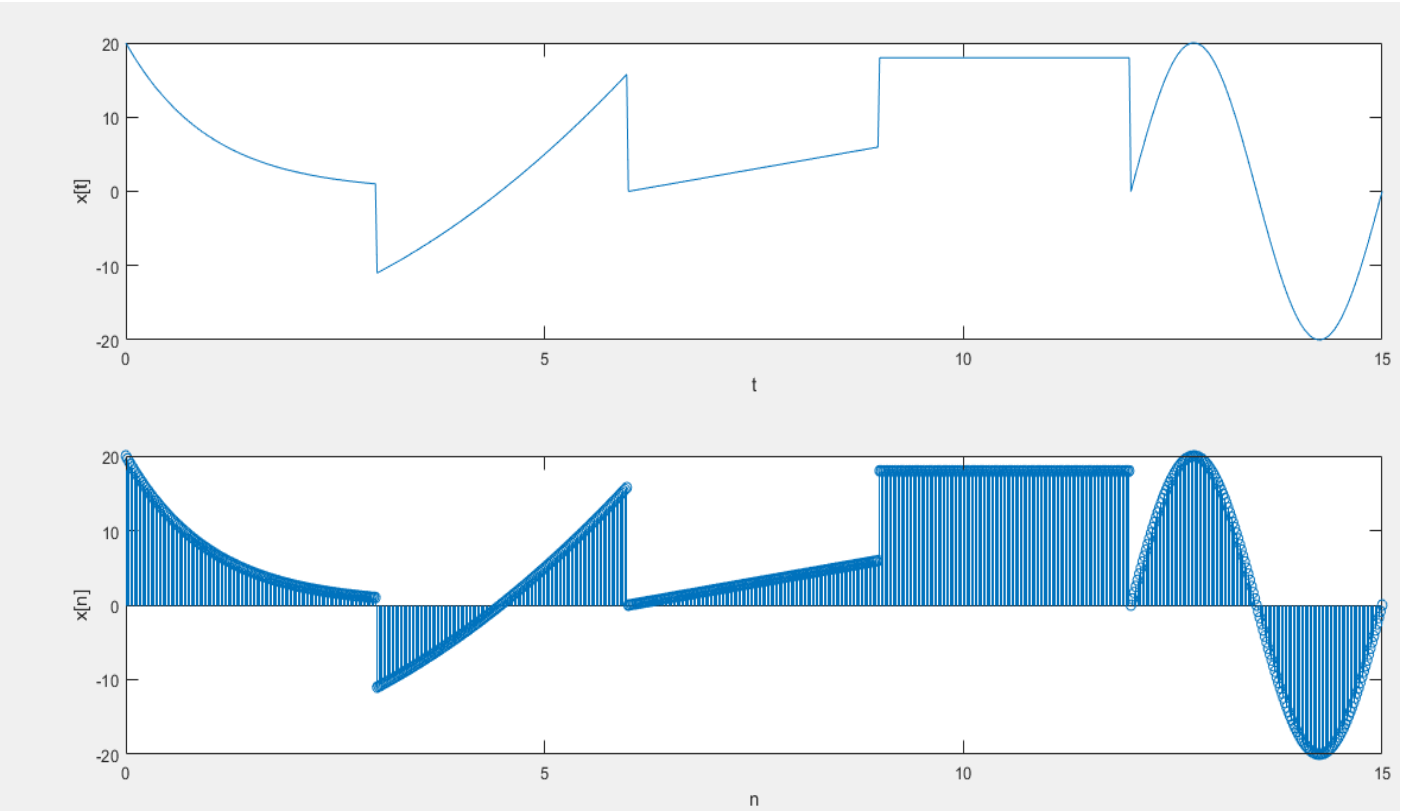


Eventually we get to the final step which is the operations that could be done on the function/signal.

Now, we show all the operations supported in this project.

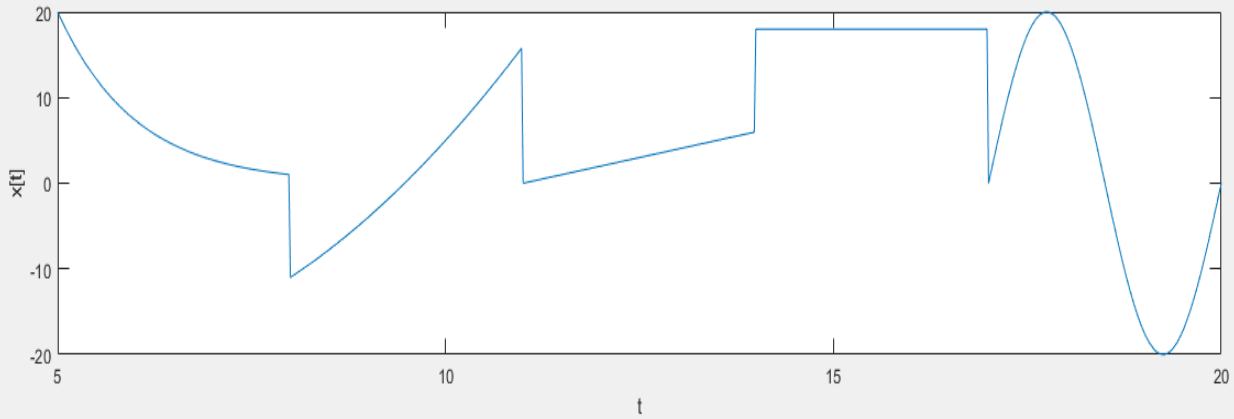
(1) Amplitude scaling

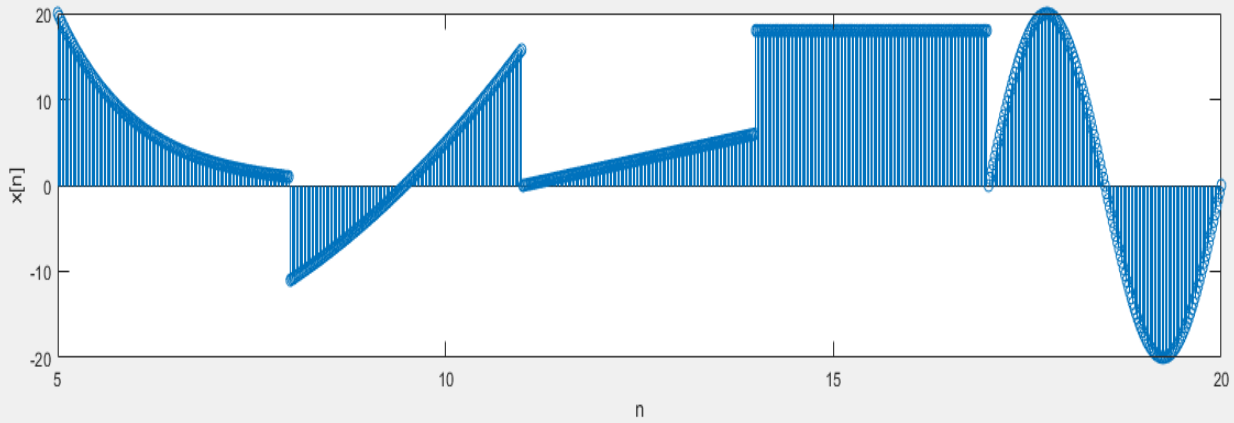
```
    ++++ Signal Operations ++++

Choose a  signal operation to be performed:
1)    Amplitude Scaling.
2)    Time Reversal.
3)    Time shift.
4)    Expanding the signal.
5)    Compressing the signal.
6)    None.
Choose [1 ~ 6]: 1
Scale Value:     2
```



(2) Time shift

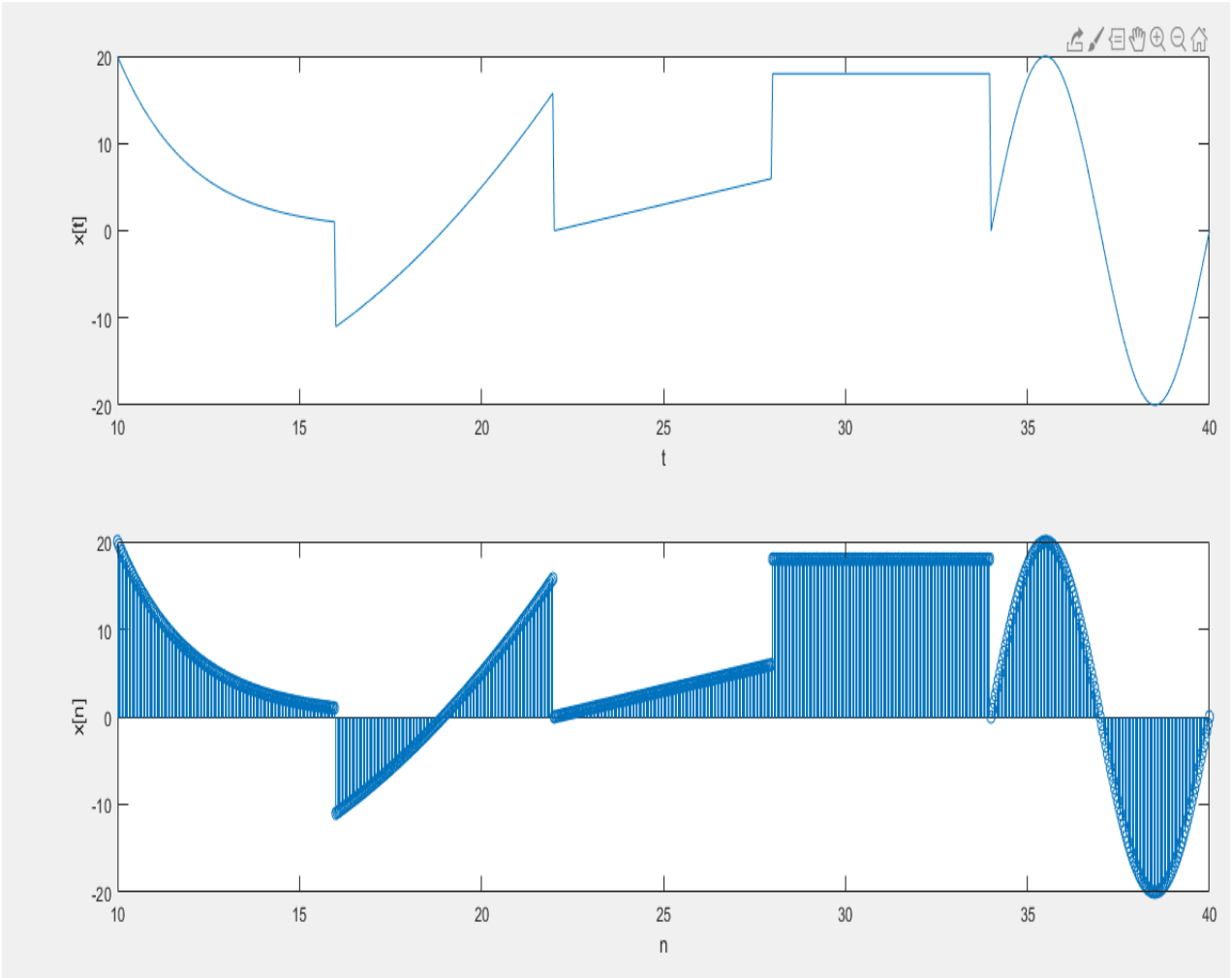
```
     ++++ Signal Operations ++++

Choose a  signal operation to be performed:
1)    Amplitude Scaling.
2)    Time Reversal.
3)    Time shift.
4)    Expanding the signal.
5)    Compressing the signal.
6)    None.
Choose [1 ~ 6]: 3
Shift Value 'X[ t - T ]':    5
```





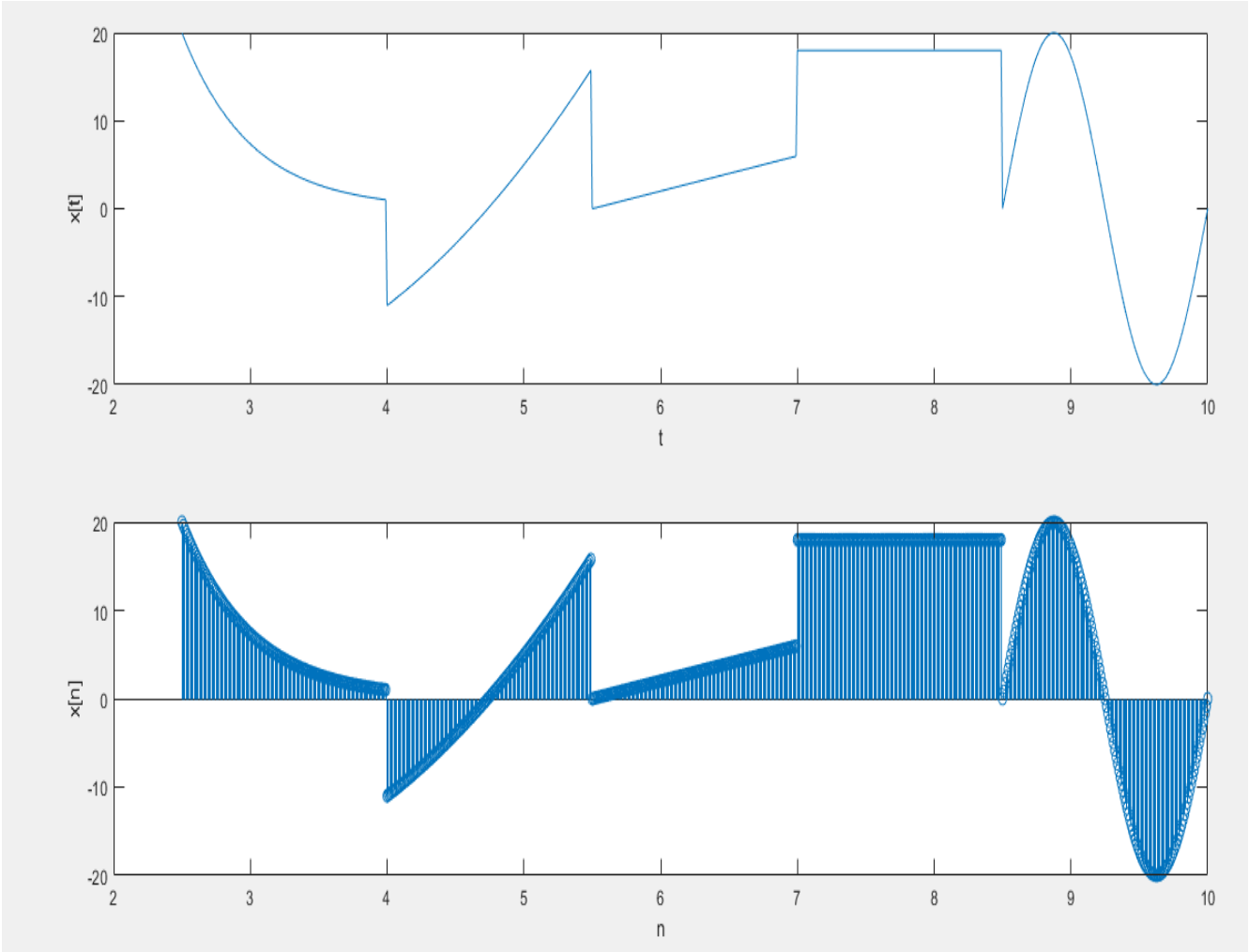
(3) Expanding the signal

```
     ++++ Signal Operations ++++

Choose a  signal operation to be performed:
1)    Amplitude Scaling.
2)    Time Reversal.
3)    Time shift.
4)    Expanding the signal.
5)    Compressing the signal.
6)    None.
Choose [1 ~ 6]: 4
Expanding Value ']0 , 1[' : 0.5
```



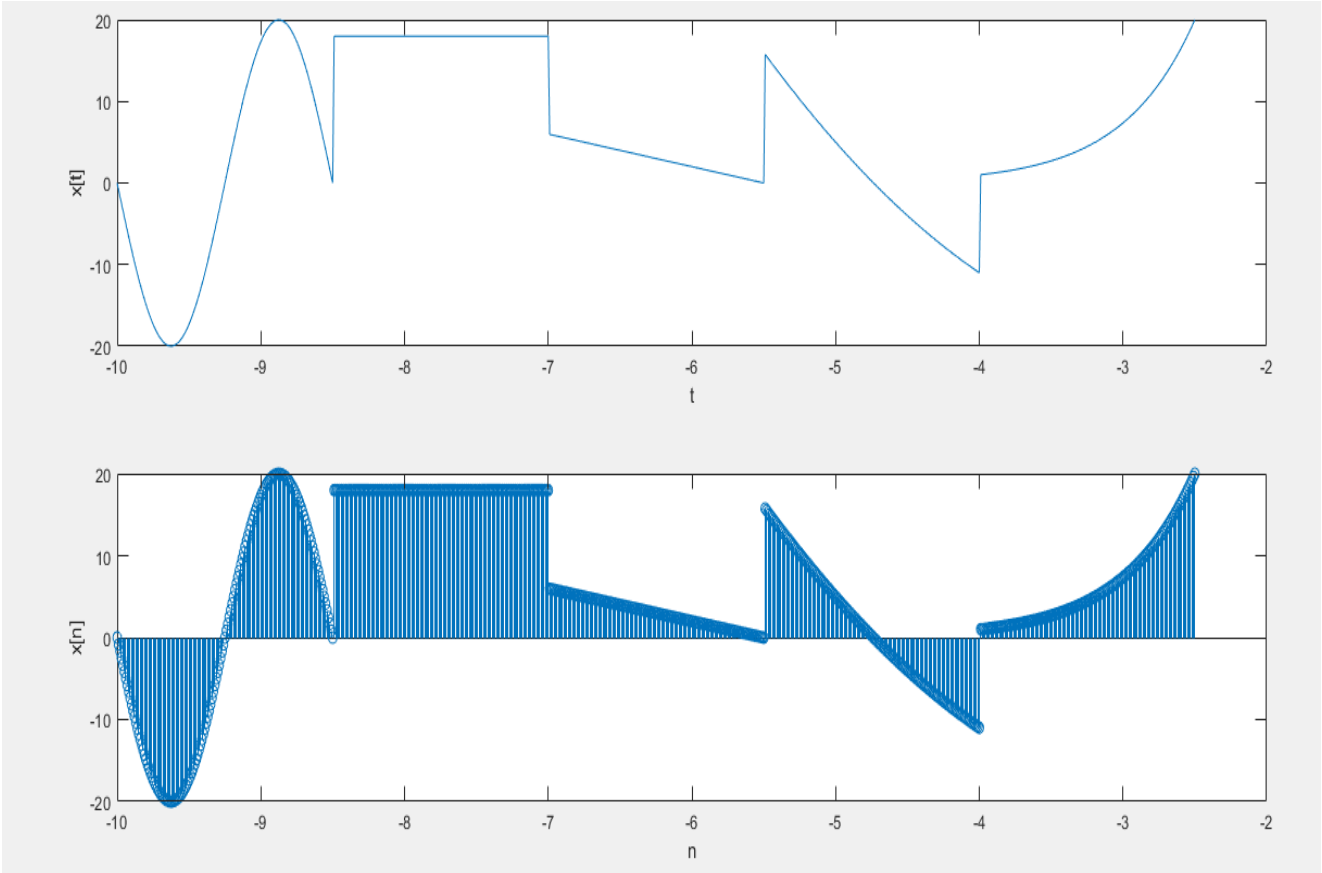
(4) Compressing the signal

```
     ++++ Signal Operations ++++

Choose a  signal operation to be performed:
1)    Amplitude Scaling.
2)    Time Reversal.
3)    Time shift.
4)    Expanding the signal.
5)    Compressing the signal.
6)    None.
Choose [1 ~ 6]: 5
Compressing Value ']1 , inf[' : 4
```



(5) Time reversal

```
    ++++ Signal Operations ++++

Choose a  signal operation to be performed:
1)    Amplitude Scaling.
2)    Time Reversal.
3)    Time shift.
4)    Expanding the signal.
5)    Compressing the signal.
6)    None.
Choose [1 ~ 6]: 2
```



Our project offers complete protection against invalid inputs.

The following screen shots show up some common invalid inputs.

(1) Frequency:

Zero & negative values are rejected:

```
Welcome to 'The General Signal Generator' project.
Define sampling frequency:    -1
        *** Invalid Input ***
Define sampling frequency:    0
        *** Invalid Input ***
Define sampling frequency:    1
```

(2) End time:

It should come later than start by at least 2 samples.

```
Define start time:        0
Define end time:          0
        *** Invalid Input ***
Define end time  :        1
        *** Invalid Input ***
Define end time  :        2
```

(3) Break points number.

It has to be integer, non-negative and below a defined max.

```
Define sampling frequency:    1
Define start time:        0
Define end time:          10
Break points number:   4
        *** Invalid Input ***
Break points number:   3
        *** Invalid Input ***
Break points number:   0.1
        *** Invalid Input ***
Break points number:   -1
        *** Invalid Input ***
Break points number:   2
```

(4) Break points' times.

They have to fit in defined regions between start and end times.

```
Break point #1 time:    0
|| Alligned to sample point at time 0
        *** Invalid Input ***
Break point #1 time:    1
|| Alligned to sample point at time 1
        *** Invalid Input ***
Break point #1 time:    2
|| Alligned to sample point at time 2
        *** Invalid Input ***
Break point #1 time:    3
|| Alligned to sample point at time 3
Break point #2 time:    10
|| Alligned to sample point at time 10
        *** Invalid Input ***
Break point #2 time:    9
|| Alligned to sample point at time 9
        *** Invalid Input ***
Break point #2 time:    8
|| Alligned to sample point at time 8
```