PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, BSK III Stage, Bengaluru-560 085
Department of Electronics and Communication Engineering

Course Title: Principles of digital signal processing
Course Code: UE21EC252B

PROJECT 1

NAME: AKASH RAVI BHAT
SRN: PES1UG21EC025

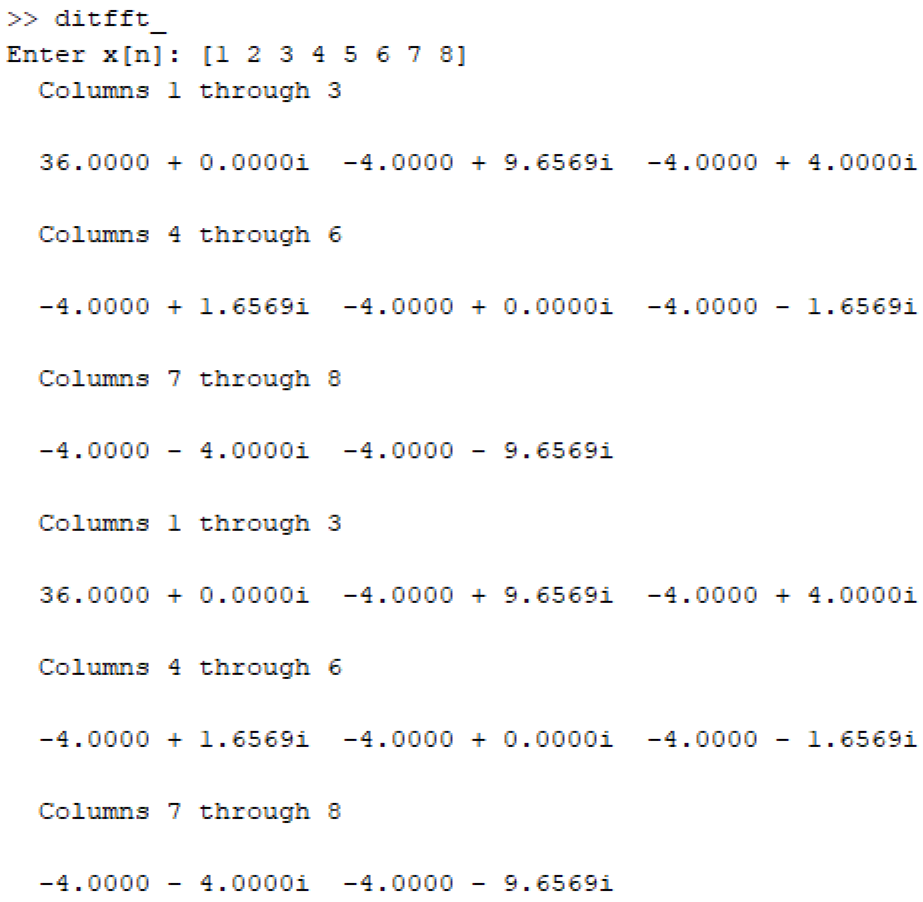
QUESTION 1:

DECIMATION IN TIME FFT

CODE:

```
%recursive (radix 2) DIT FFT
x1=input('Enter x[n]: ');% taking input
y= ditfft(x1); % calling user defined function
disp(y); % displaying output using user defined function
disp(fft(x1)) % displaying output using inbuilt function
function c = ditfft(x) % function header, c is return value
   L = length(x);
   if L > 1
       L2 = L/2;
       ce = ditfft(x(1:2:L-1)); % calling function with only even index terms;
recursive call
       co = ditfft(x(2:2:L)); % calling function with only odd index terms;
recursive call
       tf = exp(-1i*2*pi/L).^(0:L2-1); % finding twiddle factor
       c1 = tf.*co; % multiplying with twiddle factor
       c= [(ce+c1),(ce-c1)];

   else % defining base case
       c=x;
   end
end
```

OUTPUT:



```
>> ditfft_
Enter x[n]: [1 2 3 4 5 6 7 8]
  Columns 1 through 3

  36.0000 + 0.0000i  -4.0000 + 9.6569i  -4.0000 + 4.0000i

  Columns 4 through 6

  -4.0000 + 1.6569i  -4.0000 + 0.0000i  -4.0000 - 1.6569i

  Columns 7 through 8

  -4.0000 - 4.0000i  -4.0000 - 9.6569i

  Columns 1 through 3

  36.0000 + 0.0000i  -4.0000 + 9.6569i  -4.0000 + 4.0000i

  Columns 4 through 6

  -4.0000 + 1.6569i  -4.0000 + 0.0000i  -4.0000 - 1.6569i

  Columns 7 through 8

  -4.0000 - 4.0000i  -4.0000 - 9.6569i
```

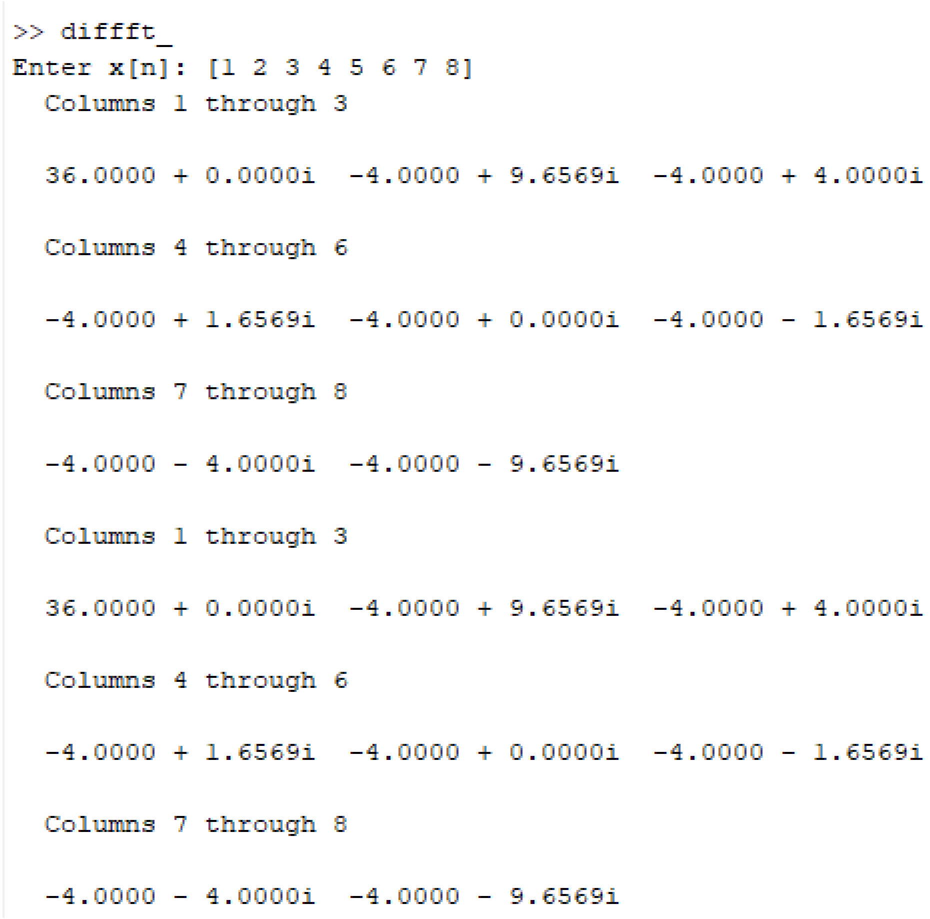
DECIMATION IN FREQUENCY FFT

CODE:

```
%recursive (radix 2) DIF FFT
x1=input('Enter x[n]: ');% taking input
y= diffft(x1); % calling user defined function
y=bitrevorder(y);% to give bit reversed input to user defined function
disp(y);% displaying output using user defined function
disp(fft(x1));% displaying output using inbuilt function
function c = diffft(x)% function header, c is return value
   L = length(x);
   if L > 1
       L2 = L/2;
       tf = exp(-1j*2*pi/L).^(0:L2-1);% finding twiddle factor
       c1 = diffft(x(1:L2)+x(L2+1:L));% recursive call
```

```
        c2 = diffft((x(1:L2)-x(L2+1:L)).*tf);% recursive call
        %cc=[c1';c2'];
        %c= cc(:);%gives first column values then 2nd column
        c=[c1,c2];

    else% defining base case
        c=x;
    end
end
```

OUTPUT:



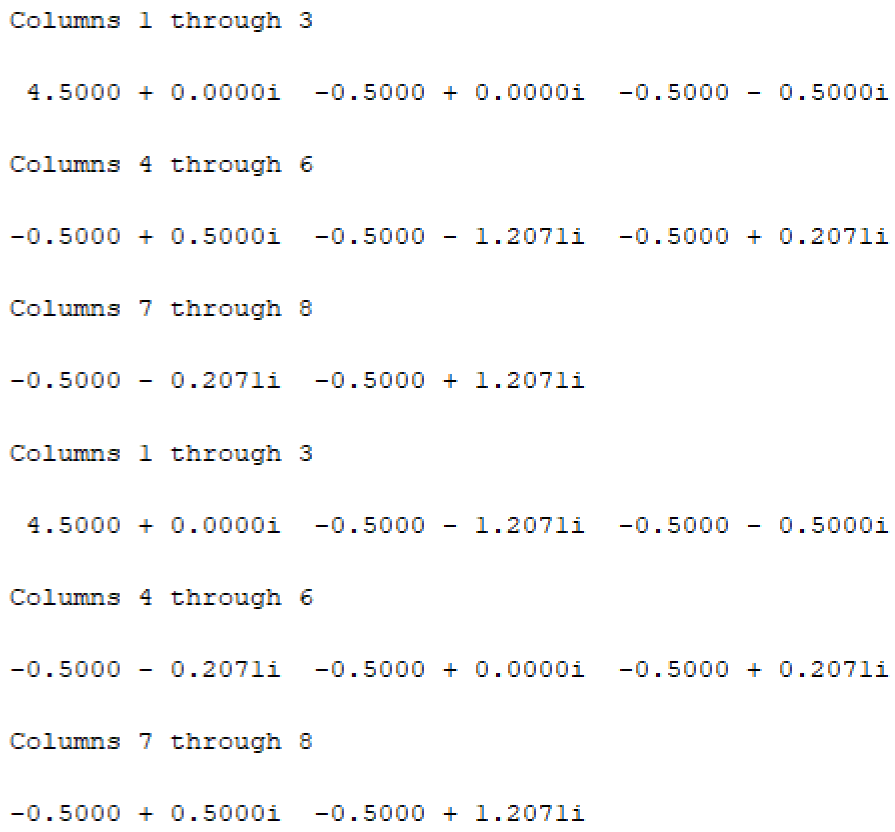
```
>> diffft_
Enter x[n]: [1 2 3 4 5 6 7 8]
  Columns 1 through 3

  36.0000 + 0.0000i  -4.0000 + 9.6569i  -4.0000 + 4.0000i

  Columns 4 through 6

  -4.0000 + 1.6569i  -4.0000 + 0.0000i  -4.0000 - 1.6569i

  Columns 7 through 8

  -4.0000 - 4.0000i  -4.0000 - 9.6569i

  Columns 1 through 3

  36.0000 + 0.0000i  -4.0000 + 9.6569i  -4.0000 + 4.0000i

  Columns 4 through 6

  -4.0000 + 1.6569i  -4.0000 + 0.0000i  -4.0000 - 1.6569i

  Columns 7 through 8

  -4.0000 - 4.0000i  -4.0000 - 9.6569i
```

DECIMATION IN TIME IFFT

CODE:

```
%recursive function for IDIT FFT
x1=input('Enter x[n]: ');% taking input
y= iditfft(x1);% calling user defined function
y=bitrevorder(y);% to give bit reversed input to user defined function
disp(y);% displaying output using user defined function
function c = iditfft(x)% function header, c is return value
   L = length(x);
   if L > 1
       L2 = L/2;
       tf = exp(1j*2*pi/L).^(0:L2-1);% finding twiddle factor
       c1 = iditfft((x(1:L2)+x(L2+1:L))./2);% recursive call
       c2 = iditfft((x(1:L2)-x(L2+1:L)).*tf./2);% recursive call
       %cc=[c1';c2'];
       %c= cc(:);%gives first column values then 2nd column
       c=[c1,c2]
   else% defining base case
       c=x;
   end
end
```

OUTPUT:



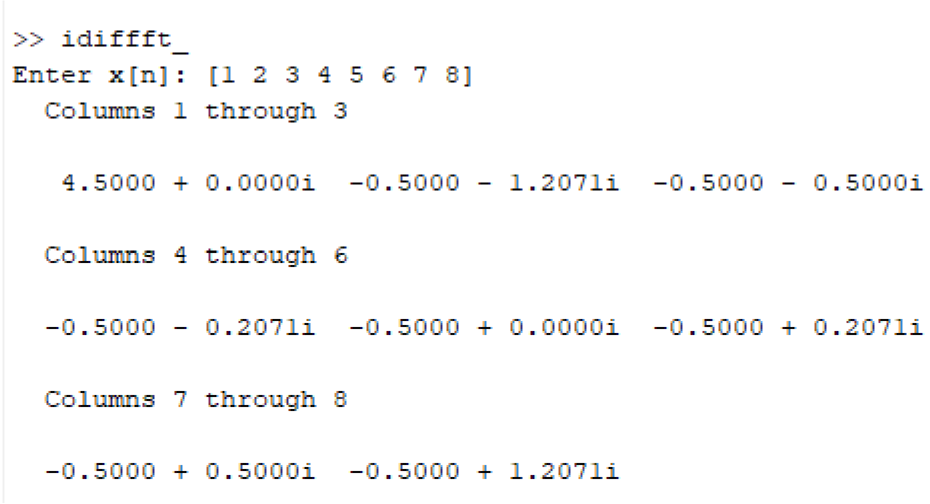
```
  Columns 1 through 3

   4.5000 + 0.0000i  -0.5000 + 0.0000i  -0.5000 - 0.5000i

  Columns 4 through 6

  -0.5000 + 0.5000i  -0.5000 - 1.2071i  -0.5000 + 0.2071i

  Columns 7 through 8

  -0.5000 - 0.2071i  -0.5000 + 1.2071i

  Columns 1 through 3

   4.5000 + 0.0000i  -0.5000 - 1.2071i  -0.5000 - 0.5000i

  Columns 4 through 6

  -0.5000 - 0.2071i  -0.5000 + 0.0000i  -0.5000 + 0.2071i

  Columns 7 through 8

  -0.5000 + 0.5000i  -0.5000 + 1.2071i
```

DECIMATION IN FREQUENCY IFFT

CODE:

```
%recursive funtion for IDIF FFT
x1=input('Enter x[n]: ');% taking input
y= ditfft(x1);% calling user defined function
disp(y);% displaying output using user defined function
function c = ditfft(x)% function header, c is return value
   L = length(x);
   if L > 1
       L2 = L/2;
       ce = ditfft(x(1:2:L-1)); % calling function with only even index terms;
recursive call
       co = ditfft(x(2:2:L)); % calling function with only odd index terms;
recursive call
       tf = exp(1i*2*pi/L).^(0:L2-1);% finding twiddle factor
       c1 = tf.*co;% multiplying with twiddle factor
       c= [(ce+c1)./2,(ce-c1)./2];% dividing by 2 at each stage

   else % defining base case
       c=x;
   end
end
```

OUTPUT:



```
>> idiffft_
Enter x[n]: [1 2 3 4 5 6 7 8]
  Columns 1 through 3

   4.5000 + 0.0000i  -0.5000 - 1.2071i  -0.5000 - 0.5000i

  Columns 4 through 6

  -0.5000 - 0.2071i  -0.5000 + 0.0000i  -0.5000 + 0.2071i

  Columns 7 through 8

  -0.5000 + 0.5000i  -0.5000 + 1.2071i
```