

Name: Shazzad Ahmed

Student ID: 220021108

Section: A2

Course No: EEE 4416

Assignment-06:

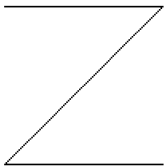
Exercise 1:

```
% part 1
ju=input('Enter number of test cases: ');
for q=1:ju
    n=input('Enter an integer : ')
    draw_z(n)
end
```

```
n = 5
ans = 5x5
    1    1    1    1    1
    0    0    0    1    0
    0    0    1    0    0
    0    1    0    0    0
    1    1    1    1    1

n = 3
ans = 3x3
    1    1    1
    0    1    0
    1    1    1
```

```
% part 2
imshow(~draw_z(100))
```



Exercise 2:

```
giu=input('Enter number of test cases: ');
for ii=1:giu
    n=input('Enter an integer : ')
    repeat_elem(n)
```

```
end
```

```
n = 3
ans = 1×6
    1    2    2    3    3    3
n = 5
ans = 1×15
    4    5    5    5    5    5
n = 1
ans = 1
```

Columns 10:15

Exercise 3:

```
nop=input('Enter number of test cases: ');
for ii=1:nop
    n=input('give integer: ')
    goldbach(n)
end
```

```
n = 10
ans = 2×2
    3    7
    5    5
n = 101
Odd numbers doesn't satisfy condition.
n = 500000
ans = 3052×2 Rows 15:24 | Columns 1:2
    379    499621
    409    499591
    541    499459
    577    499423
    673    499327
    691    499309
    733    499267
    811    499189
    859    499141
    883    499117
n = 8
ans = 1×2
    3    5
n = 20
ans = 2×2
    3    17
    7    13
```

Exercise 4:

```
calc(20, 10, "sub")
```

```
ans = 10
```

```
calc(5, 10, "div")
```

```
ans = 0.5000
```

```
calc(22, 10)
```

```
ans = 2
```

```
calc(22)
```

Not enough input arguments.

Exercise 5:

```
diag_dom([ 5, 0; 1, 5])
```

```
ans =  
'True'
```

```
diag_dom([5, 0, 0, 10; 1, 5, 5, 10; 2, 4, 4, 5; 3, 2, 2, 1])
```

```
ans =  
'False'
```

```
diag_dom([-2, 2, 1; 1, 3, 2; 1, -2, 0])
```

```
ans =  
'False'
```

```
diag_dom([-4, 2, 1; 1, 6, 2; 1, -2, 5], "strong")
```

```
ans =  
'True'
```

```
diag_dom([5, 0, 0; 1, 5, 2; 2, 4, 6], "strong")
```

```
ans =  
'False'
```

```
diag_dom([5, 0, 0; 1, 5, 2; 2, 4, 6], "weak")
```

```
ans =  
'True'
```

Exercise 6:

```
hmm=input('Enter number of test cases: ');  
for ii=1:hmm  
    a=input('Enter Array: ')  
    out=knight(a)  
end
```

```
a = 1×2  
    5    5  
out = 8×2  
    7    4  
    7    6  
    6    3  
    6    7  
    3    4  
    3    6  
    4    3  
    4    7  
a = 1×2  
    1    1
```

```

out = 2×2
     3     2
     2     3
a = 1×2
     8     8
out = 2×2
     6     7
     7     6
a = 1×2
     8     9

```

Invalid Position.

Output argument "out" (and possibly others) not assigned a value in the execution with "untitled>knight" function.

Exercise 7:

```

%Test Case - 01:
repeat_elem_v2(3)

```

```

ans = 1×6
     1     2     2     3     3     3

```

```

%Test Case - 02:
repeat_elem_v2(5,4)

```

```

ans = 1×9
     4     4     4     4     5     5     5     5     5

```

```

%Test Case - 03:
repeat_elem_v2(1)

```

```

ans = 1

```

```

%Test Case - 04:
repeat_elem_v2(1,8)

```

```

ans =

[]

```

```

%Test Case - 05:
repeat_elem_v2(6,6)

```

```

ans = 1×6
     6     6     6     6     6     6

```

Exercise 1 function:

```

function out=draw_z(n)
    if n<=2
        disp("N must be larger than 2.");
        return;
    end
    mat=zeros(n);
    mat(1,:)=1;

```

```
mat(end,:)=1;
mat(logical(flip(eye(n))))=1;
out=mat;
end
```

Exercise 2 function:

```
function rslt=repeat_elem(n)
rslt=[];
for ii=1:n
    rslt=[rslt, repmat(ii,1,ii)];
end
end
```

Exercise 3 function:

```
function ou=goldbach(n)
    if mod(n,2)~=0
        disp("Odd numbers doesn't satisfy condition.");
        return;
    end
    p=primes(n)';
    p_half=p(p<=n/2);
    complement=n-p_half;
    mask=isprime(complement);
    ou=[p_half(mask) complement(mask)];
end
```

Exercise 4 function:

```
function out=calc(a,b,op)
    if nargin<2
        disp("Not enough input arguments.");
        return;
    elseif nargin<3
        op="mod";
    end
    if op=="add"
        out=a+b;
    elseif op=="sub"
        out=a-b;
    elseif op=="mul"
        out=a*b;
    elseif op=="div"
        out=a/b;
    elseif op=="mod"
        out=mod(a,b);
    end
```

```
end
```

Exercise 5 function:

```
function yo=diag_dom(mat, op)
    if nargin<2
        op="weak";
    end
    if op=="weak"
        o=all(abs(diag(mat))>=(sum(mat,2)-diag(mat)));
    elseif op=="strong"
        o=all(abs(diag(mat))>(sum(mat,2)-diag(mat)));
    end
    if o==1
        yo='True';
    else
        yo='False';
    end
end
```

Exercise 6 function:

```
function out=knight(p)
    if ~(p(1)>=1 & p(1)<=8 & p(2)>=1 & p(2)<=8)
        disp("Invalid Position.");
        return;
    end
    offset = [+2 -1;
              +2 +1;
              +1 -2;
              +1 +2;
              -2 -1;
              -2 +1;
              -1 -2;
              -1 +2;];
    valid=p+offset;
    out=valid(all(valid>=1 & valid<=8,2),:);
end
```

Exercise 7 function:

```
function rst=repeat_elem_v2(a,b)
rst=repeat_elem(a);
if nargin>1
    if a<b
        rst=[];
    else
        rst(rst<b)=[];
    end
end
```

end
end