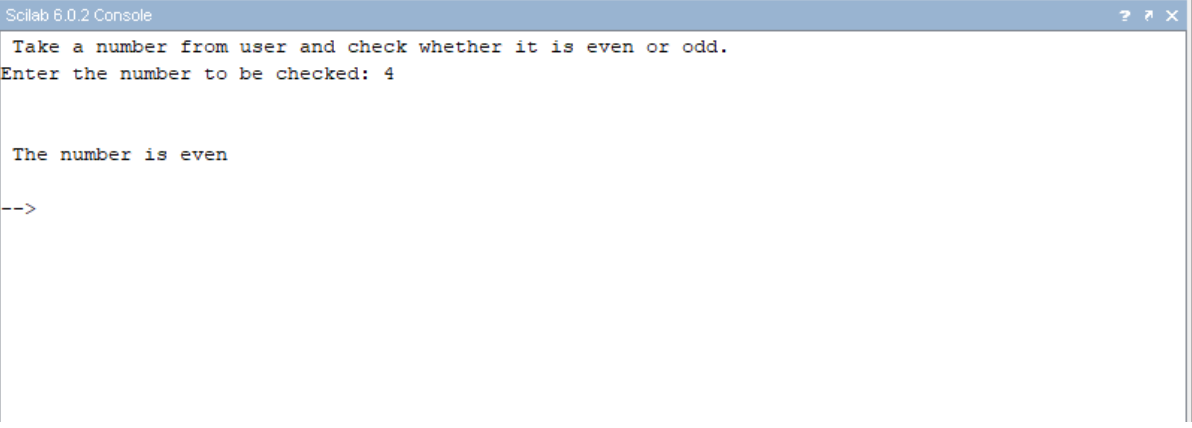|	
Name	Manish Shashikant Jadhav
UID	2023301005
Subject	Linear Algebra
Experiment No.	2
Aim	Implementation of Conditional Branching and Looping in Scilab.

1. To check whether given number is greater than 10	clc;
	printf("To check whether given number is greater than 10");
	a=5;
	disp(a);
	if a>10 then
	printf("a is more than 10");
	else
	printf("a is smaller than 10");
	end
	
	Scilab 6.0.2 Console
	To check whether given number is greater than 10
	   5.
	a is smaller than 10
	--> |

2. Write a program to check whether you are attending Linear Algebra lab for any time input from user.	clc;
	printf("Write a program to check whether you are attending Linear Algebra lab for any time input from user.");
	t=input("Enter the current time: ")
	if t>16 & t<18
	then
	printf("student is attending Linear Algebra Lab");
	else
	printf("Student has finished Linear Algebra Lab")
	end

```
Scilab 6.0.2 Console                                                    ? ↗ X
 Write a program to check whether you are attending Linear Algebra lab for any time input from user
Enter the current time: 17

student is attending Linear Algebra Lab
-->
```

3. Enter a month (number) from a user and check whether it's a first, second, third and forth quarter	clc;
	printf("Enter a month(number) from a user and check whether its a first, second, third or fourth quarter.");
	month = input("Enter a month: ")
	if month >=1 & month <=3
	then
	printf("first quarter");
	elseif month >= 4 & month <= 6
	then
	printf("second quarter");
	elseif month >= 7 & month <=9
	then
	printf("third quarter");
	elseif month >= 10 & month <=12
	then
	printf("fourth quarter");
	else
	printf("Invalid month");
	end

```
Scilab 6.0.2 Console                                                    ? ↗ X
 Enter a month(number) from a user and check whether its a first, second, third or fourth quarter.
Enter a month: 6

second quarter
-->
```
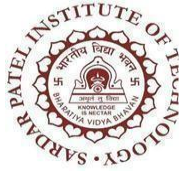
| **4.** Take a number from user and check whether it is odd or even. (Use modulo command) | ```
clc;
printf("Take a number from user and check whether it is even or odd.");
x=input("Enter the number to be checked: ")
if modulo(x,2)==0 then
disp("The number is even") ;
else
disp("The number is odd");
end
```<br><br>Scilab 6.0.2 Console     ? ↗ X<br><br> Take a number from user and check whether it is even or odd.<br>Enter the number to be checked: 4<br><br><br> The number is even<br><br>--> |
| **5.** Display values from 10 to 20 using for loop and while loop. | ```
clc;
printf("Using while loop");
a=10;
while a<=20
printf("\nValue of a: %d",a);
a=a+1;
end
``` |
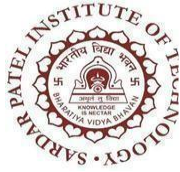
```
Scilab 6.0.2 Console                                              ? ⤢ ✕
 Using while loop
Value of a: 10
Value of a: 11
Value of a: 12
Value of a: 13
Value of a: 14
Value of a: 15
Value of a: 16
Value of a: 17
Value of a: 18
Value of a: 19
Value of a: 20
--> |
```

6. Write a program to find values for x=1 to 5 for f(x)=x^2 +sqrt(x) using for loop and while loop	clc;
	printf("Write a program to find values for x=1 to 5 for f(x)=x^2 + sqrt(x) using for and while loop.");
	for x= 1:1:5
	f(x)= x^2 + sqrt(x);
	printf('\nThe value of f(x)= %f at x= %i',f(x),x)
	end
	//using while loop
	printf("\n*****************************");
	x=1;
	while x<=5
	f(x)= x^2 + sqrt(x);
	printf('\nThe value of f(x)= %g at x= %i',f(x),x)
	x=x+1;
	end

```
Scilab 6.0.2 Console                                                    ?  ↗ X
 Write a program to find values for x=1 to 5 for f(x)=x^2 + sqrt(x) using for and while loop.
The value of f(x)= 2.000000 at x= 1
The value of f(x)= 5.414214 at x= 2
The value of f(x)= 10.732051 at x= 3
The value of f(x)= 18.000000 at x= 4
The value of f(x)= 27.236068 at x= 5
*****************************
The value of f(x)= 2 at x= 1
The value of f(x)= 5.41421 at x= 2
The value of f(x)= 10.7321 at x= 3
The value of f(x)= 18 at x= 4
The value of f(x)= 27.2361 at x= 5
--> |
```

7. Write a Scilab code to input a matrix and check whether the matrix is invertible or not.	B = [1,1 ; 2,2];
	if B == ((-B)') then
	   printf("It is Invertible \n");
	else
	   printf("It is not Invertible \n");
	end
	
	printf("\n");
	
	```Startup execution:```
	``` loading initial environment```
	
	```--> exec('D:\1.sce', -1)```
	```It is not Invertible```
	
	```-->```

8. Write a Scilab code to input a matrix and check whether the matrix is	A = [1 2+3*%i 3-4*%i; 2-3*%i 5 6+7*%i; 3+4*%i 6-7*%i 8];
	
	if A == A' then
	   disp("The input matrix is Hermitian.");
	else

Hermitian or not.	```disp("The input matrix is not Hermitian.");```
	 ```end```
	
	 ```--> exec('D:\1.sce', -1)```
	
	 ``` The input matrix is Hermitian.```
	
	 ```-->```

9. Write a Scilab code to input a matrix and check whether the matrix is Symmetric, Skew Symmetric or none.	```// Input matrix```
	 ```A = input("Enter the matrix: ");```
	
	 ```// Check if the matrix is square```
	 ```[m, n] = size(A);```
	 ```if m ~= n```
	 ``` disp('Matrix must be square');```
	 ``` return;```
	 ```end```
	
	 ```isSymmetric = isequal(A, A');```
	
	 ```isSkewSymmetric = isequal(A, -A');```
	
	 ```if isSymmetric```
	 ``` disp('Matrix type: symmetric');```
	 ```elseif isSkewSymmetric```
	 ``` disp('Matrix type: skew-symmetric');```
	 ```else```
	 ``` disp('Matrix is neither symmetric nor skew-symmetric');```
	 ```end```

```
--> exec('D:\1.sce', -1)
Enter the matrix: [1 1 -1; 1 2 0; -1 0 5]
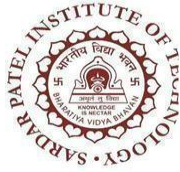

 Matrix type: symmetric


-->
```

| **10.** If Un=4(Un-1)+4 and U0=4 , Print 20th term of the Sequence | ```clc
U=[4];
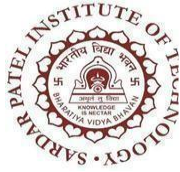for n=1:19
   U(n+1)=4*U(n)+4;
end
disp(U)
printf("\n")
printf("the 20th term is: ")
disp(U(20))
``` |

```
Scilab 6.0.2 Console                                                    ?

    4.
    20.
    84.
    340.
    1364.
    5460.
    21844.
    87380.
    349524.
    1398100.
    5592404.
    22369620.
    89478484.
    3.579D+08
    1.432D+09
    5.727D+09
    2.291D+10
    9.163D+10
    3.665D+11
    1.466D+12

the 20th term is:
    1.466D+12

-->
```

| **11.** Write a program to find the Pythagorean triplet up to the number 50 or 20. | ```
clc
for i=1:50
    for j=i+1:50
        c=(i*i)+(j*j)
        for k=i+2:50
            if c==(k*k)
                printf("\n")
                printf("Pythagorean Triplet is %d, %d, %d, \n", i,j,k);
            end
        end
    end
end
``` |

Scilab 6.0.2 Console

```
Pythagorean Triplet is 3, 4, 5,

Pythagorean Triplet is 5, 12, 13,

Pythagorean Triplet is 6, 8, 10,

Pythagorean Triplet is 7, 24, 25,

Pythagorean Triplet is 8, 15, 17,

Pythagorean Triplet is 9, 12, 15,

Pythagorean Triplet is 9, 40, 41,

Pythagorean Triplet is 10, 24, 26,

Pythagorean Triplet is 12, 16, 20,

Pythagorean Triplet is 12, 35, 37,

Pythagorean Triplet is 14, 48, 50,

Pythagorean Triplet is 15, 20, 25,

Pythagorean Triplet is 15, 36, 39,

Pythagorean Triplet is 16, 30, 34,

Pythagorean Triplet is 18, 24, 30,

Pythagorean Triplet is 20, 21, 29,

Pythagorean Triplet is 21, 28, 35,

Pythagorean Triplet is 24, 32, 40,

Pythagorean Triplet is 27, 36, 45,

Pythagorean Triplet is 30, 40, 50,
```

Conclusion	Hence, by completing this experiment I came to know about Implementation of Conditional Branching and Looping in Scilab.