|  |  |
| --- | --- |
| **Name** | Manish Shashikant Jadhav |
| **UID no.** | 2023301005 |
| **Subject** | Linear Algebra |
| **Department** | Computer Engineering-B |

|  |  |
| --- | --- |
| **Experiment 2** | |
| **AIM :** | Implementation of Conditional Branching and Looping in Scilab. |
| **Code 1:** | **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 |
| **Code 2:** | **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)) |
| **Code 3:** | **Write a Scilab code to input a matrix and check whether the matrix is Symmetric, Skew Symmetric or none.**  A = input("Enter the matrix: ");  [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 |
| **Code 4:** | **Write a Scilab code to input a matrix and check whether the matrix is Hermitian or not**  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  disp("The input matrix is not Hermitian.");  end |
| **Code 5:** | **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"); |
| **Code 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 |
| **Code 7:** | **Display values from 10 to 20 using while loop**  clc;  printf("Using while loop");  a=10;  while a<=20  printf("\nValue of a: %d",a);  a=a+1;  end |
| **Code 8:** | **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 |
| **Code 9:** | **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 |
| **Code 10:** | **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 |
| **Code 11:** | **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 |
| **CONCLUSION:** | Hence, by completing this experiment I came to know about Implementation of Conditional Branching and Looping in Scilab. |