أ. أكتب برنامج يقرأ عدد يختاره المستخدم ويرجع ما إن كان فردي أو زوجي

ب. أكتب برنامج يقرأ قيمتي متغيران يختارهما المستخدم ويتبادل قيمتهما

ت. أكتب برنامج يقرأ ثلاثة أعداد ويرتبها ترتيبا تفاضليا

ث. أكتب برنامج يولد السلسلة التالية : 0، 1، 1، 2، 3، 5، 8، 13، 21، 34، 55، 89، 144، 233، ... **حيث يحدد المستخدم طول السلسلة**

ج. برهن أن المصفوفة التالية عمودية

$$\begin{pmatrix} 1 & 2 & 0 \\ 2 & 5 & -1 \\ 4 & 10 & -1 \end{pmatrix}$$

ح. أكتب برنامج يوفر للمستخدم إمكانية رسم إحدى الدوال التالية حسب إختياره

`sin`, cos, log, exp, tan, atan, 1/x, 1/log(x), $1/x^3$

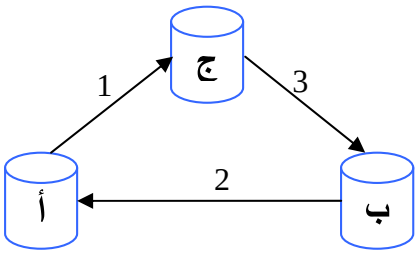
أ. أكتب برنامج يقرأ عدد يختاره المستخدم ويرجع ما إن كان فردي أو زوجي

المعروف أن الفرق الأساسي بين الأعداد الزوجية والفردية هو باقي القسمة الإقليدية للأعداد على إثنان حيث أنه يساوي صفر إذا كان العدد زوجي. إعتمادا على ذلك قمت بحساب باقي قسمت العدد (أو الرقم) المختار من قبل المستخدم ومقارنته بصفر ثم إرجاع نص إخباري بنوعيته

```
function Exercice_N1()
a=input('Please choose a number. a =')
 if (mod(a,2)==0)
    sprintf('%d is an odd number.',a)
  else
    sprintf('%d is an even number',a)
  end
end
```

ب. أكتب برنامج يقرأ قيمتي متغيران يختارهما المستخدم ويتبادل قيمتهما

من أبسط الطرق لتبادل قيمتا متغيران إستعمال متغير ثالث تخزن فيه القيم بالكيفية التالية



أ وب يمثلان المتغيران اللذان سيتم تبادل قيمتهما والمتغير ج مستعمل لخزن وقتيا قيمة المتغيرأ ليتم إسنادها للمتغير ب الذي ستعوض قبل ذلك قيمته قيمة المتغير أ

```
function Exercice_N2()
clc
 a = input('Please choose 1st number. a =');
 b = input('Please choose 2nd number. b =');
 temp=a;
 a=b;
 b=temp;
 sprintf('a = %d',a)
 sprintf('b = %d',b)
end
```

ت. أكتب برنامج يقرأ ثلاثة أعداد ويرتبها ترتيبا تفاضليا

يمكن حل هذا التمرين بعدة طرق سأبرز طريقتين منها إحداهما عبر كتابة خوارزمية متكاملة لحل الإشكال والطريقة الثانية عبر إستعمال الدوال التي يوفرها البرنامج

إثر قراءة قيم المتغيرات الثلاثة تتم مقارنتها أزواجا وطباعة الترتيب في كل حالة. الكود التالي يجسد كيفية عمل خوارزمية الترتيب

```
function Exercice_N3V0()
 a = input('Please choose 1st number. a =');
 b = input('Please choose 2nd number. b =');
 c = input('Please choose 3rd number. c =');

  if (a<b)
    if (b<c)
        sprintf('%d < %d < %d',a ,b ,c )
    else if (a<c) % a<b & c<b
        sprintf('%d < %d < %d',a ,c ,b )
        else sprintf('%d < %d < %d',c ,a ,b )
        end
    end
  else
    if (b>c)
        sprintf('%d < %d < %d',c ,b ,a )
    else if(c>a)  %c>b & a>b
            sprintf('%d < %d < %d',b ,a ,c )
        else % a>b & a>c
           sprintf('%d < %d < %d',b ,c ,a )
        end
    end
  end
end
```

إثر كتابة الخوارزمية والتأكد من سلامة عملها قمت بإضافة دالة خارجية تتكفل بوضيفة طباعة نص النتيجة في شاشة التعليمات وذلك لتجنب إعادة كتابة التعليمات المعنية و التعود على كتابة الدوال والتصرف فيها

```
function Exercice_N3V1()
 a = input('Please choose 1st number. a =');
 b = input('Please choose 2nd number. b =');
 c = input('Please choose 3rd number. c =');

  if (a<b)
    if (b<c)
        Show(a ,b ,c )
     else if (a<c) % a<b && c<b
           Show(a ,c ,b )
           else Show(c ,a ,b )
          end
     end
  else
    if (b>c)
        Show(c ,b ,a )
     else if(c>a)  %(c>b) & a>b
            Show(b ,a ,c )
          else %a>b & a>c
           Show(b ,c ,a )
         end
    end
  end
end
```

```
function Show(a,b,c)
 sprintf('%d < %d < %d',a ,b ,c )
end
```

الطريقة الثانية وهي أبسط من الأولى لأن الدالة ***sort*** التي يوفرها البرنامج توفر الكثير من الوقت والجهد وتعوض الخوارزمية السابقة. هذه الدالة ترتب عناصر المصفوفة الأحادية البعد ترتيبا تصاعديا

```
function Exercice_N3V1()
  a = input('Please choose 1st number. a =');
  b = input('Please choose 2nd number. b =');
  c = input('Please choose 3rd number. c =');
  Vector =[a,b,c];
  Sorted_Vector = sort(Vector);
  Show(Sorted_Vector(1),
       Sorted_Vector(2),
       Sorted_Vector(3))
end
```

ث. أكتب برنامج يولد السلسلة التالية : 0، 1، 1، 2، 3، 5، 8، 13، 21، 34، 55، 89، 144، 233، ... حيث يحدد المستخدم طول السلسلة

نلاحظ أن العناصر المكونة للسلسلة مترابطة مع بعضها حيث أن قيمة كل عنصر تساوي مجموع العنصرين السابقين. لذلك قمت بحجز مصفوفة أحادية الأبعاد يتكون عناصرها من الأرقام 0,1,2,3 ... إلى حدود الرقم الذي يختاره المستخدم وبهذه الطريقة يتم توفير الرقمين الأولين بشكل تلقائي وإعتماد عليهما يتم حساب البقية

```
function Exercice_N4()
 max=input('Please choose the series length: ');
 Vector = [0:max];
   for i=3:max+1
      Vector(i) = Vector(i-1) + Vector(i-2);
   end
 Vector
end
```

ج. برهن أن المصفوفة التالية عمودية

$$\begin{pmatrix} 1 & 2 & 0 \\ 2 & 5 & -1 \\ 4 & 10 & -1 \end{pmatrix}$$

المصفوفة العمودية هي المصفوفة التي يكون ناتج ضربها في مقلوبها يساوي المصفوفة التالية

$$\begin{pmatrix} 1 & 0 & 0 \dots & 0 \\ 0 & 1 & 0 \dots & 0 \\ 0 & 0 & 1 & \\ : & : & : & \\ 0 & 0 & 0 \dots & 1 \end{pmatrix}$$

في برنامج هذا التطبيق أضفت تعليمات إضافية لتيسير التعامل مع المصفوفات ذات البعدين و تجسيد جانب نظري للتعامل مع المصفوفات

```
function Exercice_N5()
A = [1 2 0; 2 5 -1; 4 10 -1]
Transpose_A= A'
Multiplication_1= A*A'
Multiplication_2= A.*A'
Inverse_A= inv(A)
I=inv(A)*A

Eigenvalues_A = eig(A)
singular_value_decomposition=svd(A)

Characterisitic_polynomial_A = round(poly(A))
eig_original=roots(Characterisitic_polynomial_A)

Convolution=conv(Characterisitic_polynomial_A,
                 Characterisitic_polynomial_A)

result = conv(Characterisitic_polynomial_A,
              Convolution)
plot(result)
title('conv(p,conv(p,p)) where p=round(poly(A))')
grid
end
```

ج. أكتب برنامج يوفر للمستخدم إمكانية رسم إحدى الدوال التالية حسب إختياره

sin, cos, log, exp, tan, atan, 1/x, 1/log(x), $1/x^3$

```
function Exercice_N6()
syms x;
sprintf('Please choose a function to plot\n \t1.
sin\n\t2. cos\n\t3. log\n\t4. exp\n\t5. tan\n\t6.
atan\n\t7. 1/x\n\t8. 1/log\n\t9. 1/x^3')
choice = input('Choose function to plot:');
 switch(choice)
    case 1, f = sin(x);
    case 2, f = cos(x);
    case 3, f = log(x);
    case 4, f = exp(x);
    case 5, f = tan(x);
    case 6, f = atan(x);
    case 7, f = 1/x;
    case 8, f = 1/log(x);
    case 9, f = 1/(x^3);
  end
 ezplot(f)
 grid
end
```

مع إضافة نص مساعدة .هذا الكود الرئيسي للبرنامج وفيه قمت بالتوفير للمستخدم حرية إختيار التطبيق الذي يريد تجربته لتيسير عملية الإختيار

```
function Exercice_Solutions()
clc
sprintf('\n\t1. Exercice N°1\n\t2. Exercice N°2\n\t3. Exercice
N°3\n\t4. Exercice N°4\n\t5. Exercice N°5\n\t6. Exercice N°7\n')

choice=input('Please select Exercice Nbr to check his solution:');
 switch(choice)
    case 1, Exercice_N1();
    case 2, Exercice_N2();
    case 3, Exercice_N3V0;
    case 4, Exercice_N4();
    case 5, Exercice_N5();
    case 6, Exercice_N6();
 end
end
```