Q3) Critical Points of a function (3 marks)

i) Generate a third degree polynomial in x and y named g(x, y) that is based on your mobile number (Note : In case there is a 0 in one of the digits replace it by 3). Suppose your mobile number is 9412821233, then the polynomial would be $g(x, y) = 9x^3 - 4x^2y + 1xy^2 - 2y^3 + 8x^2 - 2xy + y^2 - 2x + 3y - 3$, where alternate positive and negative sign are used. Deliverable(s) : The polynomial constructed should be reported. (0.5)

Mobile number: 9545521116

```
var('x,y')
g(x,y)=9*x^3 - 5*x^2*y + 4*x*y^2- 5*y^3+ 5*x^2- 2*x*y + y^2 - x + y - 6
print("Polynoimal is:", g(x,y))
```

Ans:

Polynomial is: 9*x^3 - 5*x^2*y + 4*x*y^2 - 5*y^3 + 5*x^2 - 2*x*y + y^2 - x + y - 6

ii) Write a code to find all critical points of g(x, y). You may use built in functions like ’solve’ (or other similar functions) in Octave/Matlab to find the critical points . Deliverable(s) : The code that finds the critical points along with the display of all the calculated critical points. (1)

Answer in the below screenshot

iii) Write a code to determine whether they correspond to a maximum, minimum or a saddle point. Deliverable(s) : The code that identifies the type of critical points. The critical points and their type must be presented in the form of the table generated by code for the above polynomial. (1.5 marks)

```
var('x,y')

g(x,y)=9*x^3 - 5*x^2*y + 4*x*y^2- 5*y^3+ 5*x^2- 2*x*y + y^2 - x + y - 6

print("Polynomial is:", g(x,y))

# Fermat's theorem and second derivative test

gx(x,y)=diff(g(x,y),x)

gy(x,y)=diff(g(x,y),y)

gxy(x,y)=diff(gx,y)

gxx(x,y)=diff(gx, x)

gyy(x,y)=diff(gy,y)

cpts= solve([gx,gy],x,y)

h(x,y)=gxx*gyy-gxy*gxy

print("gx=",gx(x,y))

print("gy=",gy(x,y))
```

```
print("critical points of f are:",cpts)
print("gxy=",gxy(x,y))
print("gxx=", gxx(x,y))
print("gyy=",gyy(x,y))
print("hessianf=",h(x,y))
# extracting the real roots of the given polynoimal
a1= cpts[0][0]
b1=cpts[0][1]
c1=cpts[1][0]
d1=cpts[1][1]
print(a1)
print(b1)
print(c1)
print(d1)
print("real critical points of f are [a1,b1] and [c1,d1]:",[a1.rhs(),b1.rhs()],[c1.rhs(),d1.rhs()])
t=simplify(h(a1,b1)).rhs()
s=simplify(h(c1,d1)).rhs()
print("hessian of g at (a1,b1) is :", t)
print("hessian of g at (c1,d1) is :",s)
# to find the value of gxx*gyy- gxy*gxy
A= simplify(gx(a1,b1)).rhs()
B= simplify(gy(a1,b1)).rhs()
C= simplify(gxy(a1,b1)*gxy(a1,b1)).rhs()
k= simplify(g(a1,b1)).rhs()
# to identify the local extrema at the real roots of g at the critical points (a1,b1)
f ((A < 0) and ( C> 0)):
  print( "g has local minimum at (a1,b1) and lmin value is:",k)
else:
   if ((A > 0) and (C >0 )):
     print("g has local maxium at (a1,b1), lmax value is:",k)
   else:
```

```
    if((A==0)and (B==0)):
     if(C<0):
          print("g has saddle point at(c1,d1):", K, L, M)
    else:
      if (C < 0):
        print("g has inflection point at (a1,b1):",k)
      else:
          if ( C== 0):
            print("no method to conclude:",k)
 # to identify the local extrema at the real roots of f at the critical points (c,d)
# to find the value of fxx*fyy- fxy*fxy
K= simplify(gx(c1,d1)).rhs()
L= simplify(gy(a1,b1)).rhs()
M= simplify(gxy(a1,b1)*gxy(a1,b1)).rhs()
v= simplify(g(c1,d1)).rhs()
if ((K < 0) and ( M> 0)):
  print( "g has local minimum at (c1,d1) and lmin value is:",v)
else:
  if ((K > 0) and (M >0 )):
    print("g has local maximum at (c1,d1), lmax value is:",v)
  else:
    if((K==0) and (L==0)):
      if(M<0):
         print("g has saddle point at(c1,d1):", K, L, M)
    else:
       if (M < 0):
        print("g has inflection point at (c1,d1):",v)
       else:
        if ( M== 0):
          print("no method to conclude:",v)
```

```
Polynomial is: 9*x^3 - 5*x^2*y + 4*x*y^2 - 5*y^3 + 5*x^2 - 2*x*y + y^2 - x + y - 6
gx= 27*x^2 - 10*x*y + 4*y^2 + 10*x - 2*y - 1
gy= -5*x^2 + 8*x*y - 15*y^2 - 2*x + 2*y + 1
critical points of f are: [
[x == 0.03925493950735486, y == -0.1814553269880258],
[x == 0.1225683255758444, y == 0.3342713096494314],
[x == -0.403444929645803, y == 0.2196148748733141],
[x == -0.4845768520959662, y == -0.3011294526498697]
]
gxy= -10*x + 8*y - 2
gxx= 54*x - 10*y + 10
gyy= 8*x - 30*y + 2
hessianf= -4*(5*x - 4*y + 1)^2 + 4*(27*x - 5*y + 5)*(4*x - 15*y + 1)
x == 0.03925493950735486
y == -0.1814553269880258
x == 0.1225683255758444
y == 0.3342713096494314
real critical points of f are [a1,b1] and [c1,d1]: [0.03925493950735486, -0.1814553269880258] [0.1225683255758444, 0.3342713096494314]
hessian of g at (a1,b1) is : 93.32045267624268
hessian of g at (c1,d1) is : -93.86784110238254
g has local maxium at (a1,b1), lmax value is: -6.128847905264392
g has local minimum at (c1,d1) and lmin value is: -5.823894638351256
```