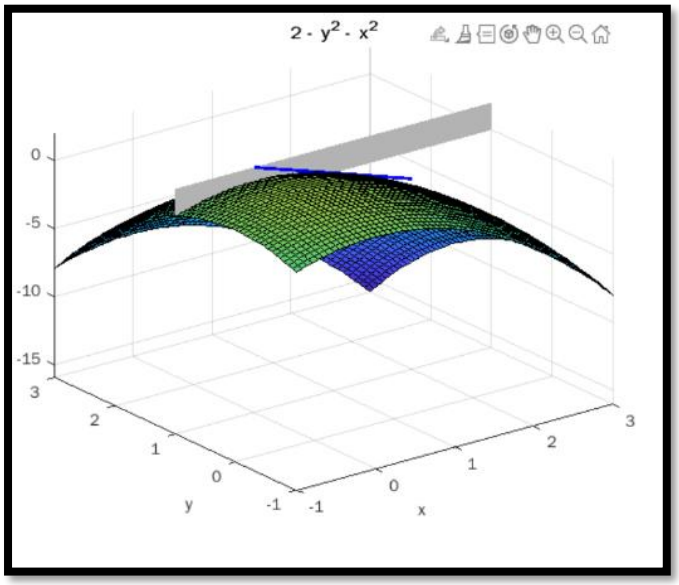
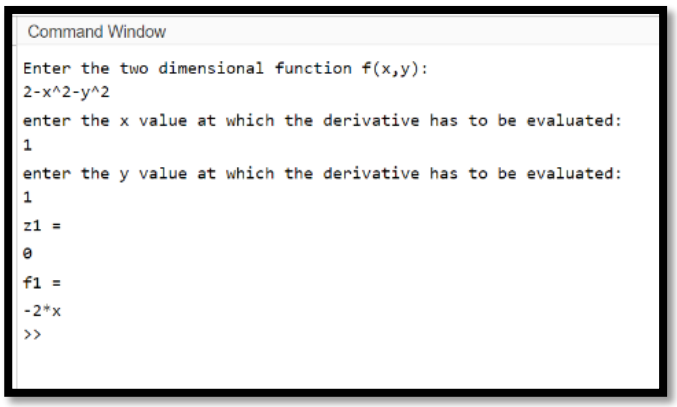
Q1:

```
clc
clear all
format compact
syms x y
z = input('Enter the two dimensional function f(x,y): ');
x1 = input('enter the x value at which the derivative has to be evaluated: ');
y1 = input('enter the y value at which the derivative has to be evaluated: ');

z1 = subs(subs(z,x,x1),y,y1)
ezsurf(z,[x1-2 x1+2])
f1 = diff(z,x)
slopex = subs(subs(f1,x,x1),y,y1);
[x2,z2]=meshgrid(x1-2:.25:x1+2,0:0.5:10);
y2=y1*ones(size(x2));
hold on
h1=surf(x2,y2,z2);
set(h1,'FaceColor',[0.7,0.7,0.7],'EdgeColor','none')
t=linspace(-1,1);
x3=x1+t;
y3=y1*ones(size(t));
z3=z1+slopex*t;
line(x3,y3,z3,'color','blue','linewidth',2)
```

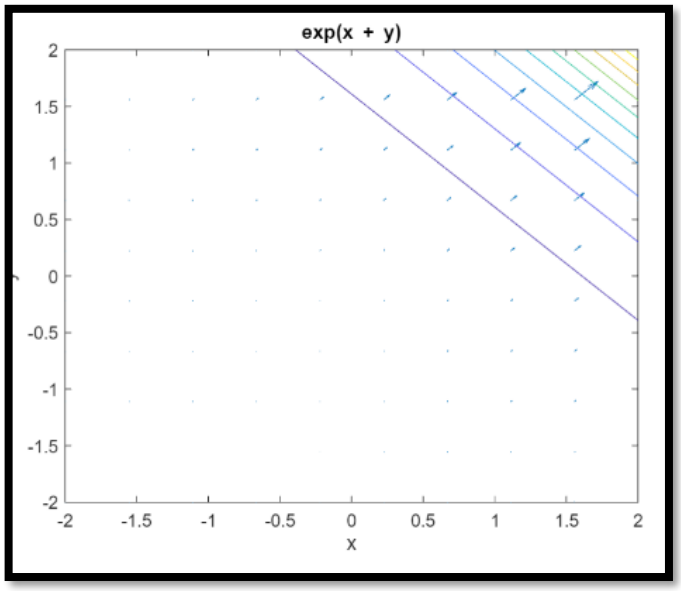




Q2:

```
clc
clear all
syms x y
f=input('enter the function f(x,y):');
F=gradient(f)
P = inline(vectorize(F(1)), 'x', 'y');
Q = inline(vectorize(F(2)), 'x','y');
x = linspace(-2, 2, 10);
y = x;
[X,Y] = meshgrid(x,y);
U = P(X,Y);
V = Q(X,Y);
quiver(X,Y,U,V,1)
axis on
xlabel('x')
ylabel('y')
hold on
ezcontour(f,[-2 2])
```

a)



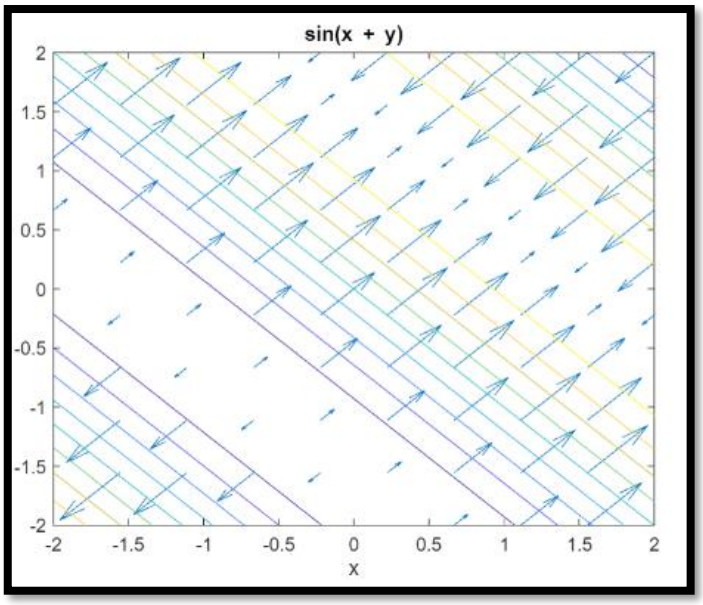

```
enter the function f(x,y):
>> exp(x+y)

F =

exp(x + y)

exp(x + y)
>>
```

b)




```
enter the function f(x,y):
sin(x+y)
F =
cos(x + y)
cos(x + y)
>>
```

Q3:

Debug:

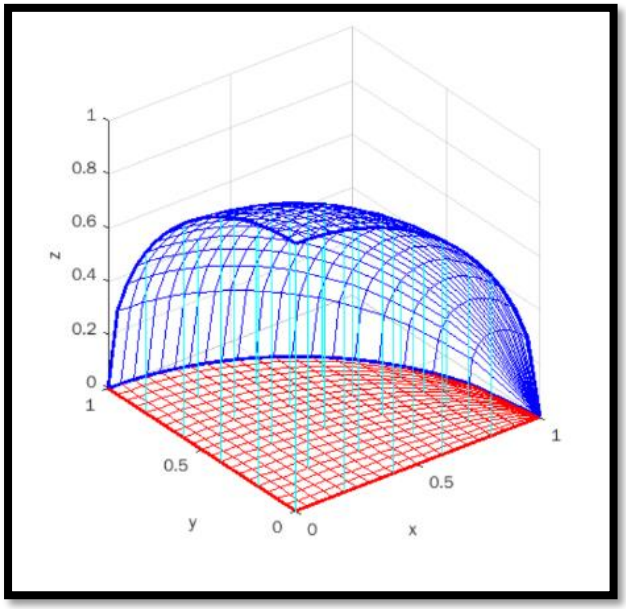
```
clc
clear all
syms x y z
I=int(int(sqrt(1-x^2-y^2),y,0,sqrt(1-x^2)),x,0,1)

viewSolid(z,0+0*x*y,sqrt(1-x^2-y^2),y,0+0*x,sqrt(1-x^2),x,0,1);

axis equal; grid on
axis on
xlabel('x')
ylabel('y')
zlabel('z')
```

Original:

```
syms x y z
I=int(int(sqrt(1-x^2-y^2), y,0, sqrt(1-x^2)), x,0,1)
viewSolid (z,0+0*x*y,4-2*x-2*y, y,0+0*x,2-x, x,0,2) ;
axis equal; grid on
axis on
xlabel('x')
ylabel('y')
zlabel('z'
```




```
I =

pi/6
```

Q4:

DEBUG:

```
clc
clear all
syms x
f=input('enter the function f(x):');
a=input('enter lower limit of x ');
b=input('enter the upper limit of x');
n=input('number of intervals');
z=int(f,a,b)
value = 0;
dx = (b-a)/n;
for k=1:n
    c = a+k*dx;
    d=subs(f,x,c);
    value = value + c;
end
value = dx*value
ezplot(f)
rsums(f, a, b)
```

Original:

```
syms x
f=input('give the function')
a=input('lower limit')
b=input('upper limit')
n=input('number of intervals')
z=int(f,a,b)
value=0;
dx=(b-a)/n;
for k=1:n
   c=a+k*dx;
   d=subs(f,x,c);
   value=value+c
end
value=dx*value
ezplot(f,[a b])
z=int(f,a,b)
rsums(f,a,b)
```