NAME: ....

STUDENT NUMBER: ....

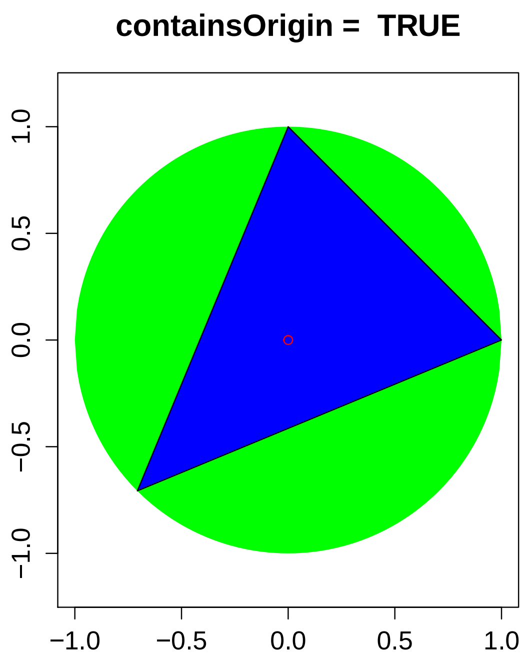
Note that if your solutions are **not functional** then they must be **dysfunctional**.

Question 1 (10 marks)

Consider three points on the circumference of the unit circle. Construct a function called `containsOrigin` that decides whether or not the origin of the unit circle is enclosed by the triangle defined by the three points.

Your function should take a vector of length three as input and return a boolean. Each element of the input vector should be an angle, θ_i, measured in radians in the range $0 \leq \theta_i < 2\pi$, defining a point on the unit circle.

For example, the vector `c( 0 , pi/2 , 5*pi/4 )` defines three points on the unit circle which in turn define the triangle shown in the figure below which in this case does contain the origin of the unit circle.



Note: use a comment statement in your code to describe the decision your function will make in the event that the origin is exactly on the boundary of the triangle.

write your question 1 solution on this page

```
########## Question 1, containsOrigin

containsOrigin <- function(thetas=c(0, pi/2, 5*pi/8))
{
  # make sure each angle is in the range 0 .. 2pi and order them
  t <- sort(thetas %% ( 2 * pi ))
  # rotate so that t[1] is zero
  t <- t - t[1]
  # check for containment with boundary condition returning TRUE
  if ( (t[2] <= pi) && (t[3] >= pi) &&
       (t[3] <= t[2] + pi) ) {
    return(TRUE)
  } else {
    return(FALSE)
  }
}
```

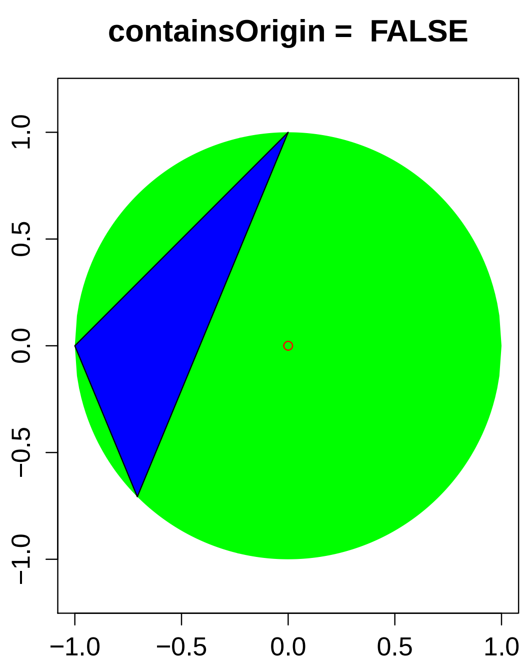
Question 2 (15 marks)

Construct an R function called `containsOriginPlot` that takes the same input vector as described in question 1 and produces a plot of the unit circle with the input triangle inscribed. Your plot should be annotated with a heading that announces whether or not the origin is contained within the triangle.

For example a call of:

```
containsOriginPlot( c( pi/2 , pi , 5*pi/4 ) )
```

should produce the figure shown below.



write your question 2 solution on this page

```
########## Question 2, containsOriginPlot

containsOriginPlot <- function(thetas=c(0, pi/2, 5*pi/4))
{

  # create a blank sheet
  plot(x=c(-1,1),y=c(-1,1), xlim = c(-1,1), ylim=c(-1,1),
       type='n', asp = 1, xlab="", ylab="")

  # generate many points on the unit circle
  x <- seq(-1, 1, 0.01)
  y <- sqrt( 1 - x^2)
  x <- c(x,rev(x))
  y <- c(y,-rev(y))

  # draw the circle
  polygon(x,y, col = 'green', border = NA)

  # convert the three angle to (x,y) coords
  # and draw the triangle
  polygon(x=cos(thetas),y=sin(thetas), col = 'blue')

  # draw the origin
  points(x=c(0),y=c(0),col = 'red')

  # add a title announcing the decision of your
  # containsOrigin function
  title(main=paste("containsOrigin = ",
                   containsOrigin(thetas)))

}

# an example for the figure in the test question
containsOriginPlot( c(pi/2, pi, 5*pi/4) )

# example plot for a random triangle
containsOriginPlot(thetas = 2 * pi * runif(3))
```

Question 3 (10 marks)

Making use of your function from question 1 above, construct an R script that will calculate the probability that a random triangle inscribed on the unit circle will contain the origin. Write your script in the space below:

```
########## Question 3, containsOriginProbability

# a function to compute the probability of a random triangle
# containing the origin from n trials (with default n=10000)

containsOriginProbability <- function (n=10000)
{
  # generate 3*n random angles
  ta <- 2 * pi * runif(3*n)
  # partition the random triangles into n sets of 3
  triangles <- lapply(1:n, function(i){c(ta[(3*i-2):(3*i)])})
  # apply the containsOrigin ftn to each triangle and
  # add up the TRUEs
  contains <- sum(sapply(triangles,containsOrigin))
  # return the probability as the ratio of TRUE's to
  # the total no of trials
  return(contains/n)
}

# a test run
containsOriginProbability()

########## Question 3, with data.frame

# the same thing using a data.frame
n <- 10000
df <- data.frame(matrix(2*pi*runif(n*3),ncol=3))
names(df) <- c("t1","t2","t3")
df$origin <- apply(df, 1,
     function(d) containsOrigin(c(d["t1"],d["t2"],d["t3"])) )
(prob <- sum(df$origin) / nrow(df))
```

Homework (tear off this page and take it home with you)

The inspiration for these test questions came from a Gilbert Strang lecture where the famous professor discusses the probability of **any** random triangle on the plane being an ıacute triangle. A triangle is *acute* if all three internal angles are less than $\frac{\pi}{2}$.

In this lecture he mentions the problem of computing the probability that any random triangle inscribed on the unit circle is an acute triangle and he suggests that this computation probably has the same result but he is not sure.

These scripts you have constructed in this test allow you to answer this question via simulation.

For homework view the webinar and then try and derive an exact expression for the probability of any random triangle on the unit sphere being acute.

The webinar may be viewed here:

`http://math.mit.edu/~gs/videos/index.html`

Your solution may involve integrals in which case you may make use of *Mathematica* to do the integration for you.

This homework problem is not for marks so don't panic if you can't solve it.

On the other hand if you do manage to solve it then feel free to email me your solution.