Coding :

```r
int.est = function(x, conf){

  # Calculate the area for the confidence interval
  area = (conf / 100) + ((1 - (conf / 100)) / 2)

  # Calculate the standard error (se) of the data
  se = sd(x) /  sqrt(length(x))

  # Calculate the margin of error using the t-distribution and specified confidence area
  margin = qt(area, length(x) - 1) * se

  # Calculate the lower and upper bounds of the confidence interval
  lower = mean(x) - margin
  upper = mean(x) + margin

  cat("lower = ", lower, " upper = ", upper)

}

score = c(83, 73, 62, 63, 71, 77, 77, 59, 92)
int.est(score, 95)

t.test(score)$conf.int
```

Result :

```
> int.est(score, 95)
lower =  64.78388  upper =  81.21612>
> t.test(score)$conf.int
[1] 64.78388 81.21612
attr(,"conf.level")
[1] 0.95
```

Conclusion :

The output of both the int.est function and the t-test are compared and found to be identical, demonstrating that both methods produce the same confidence interval for the mean of the data. This is due to t.test's default confidence intervals being 95%, and our function also accepted 95 as the parameter for confidence interval. When you have a small amount of data less than 40 sample size (in our case we only have 9), it's a good idea to choose a higher confidence level to make your estimate more reliable, because the small sample size makes the estimate less accurate.