

ACTIVIDADES.

ACT.1.-

```
prueba<-function(a=3, b=7, c=6)
{
  j="joel_alberto"
  y=a+b*c
  print(y)
  print(j)
}
prueba()
```

ACT.2.-

```
prueba<-function(a=3, b=7, c=6)
{
  j="joel_alberto"
  y=a+b*c
  print(y)
  print(j)
}
```

```
prueba2<-function()
{
  a1<-1
  b1<-1
  c1<-1
  h<-0
  for(h in 1:5)
  {
    prueba(a1,b1,c1)
    a1=a1+1
    b1=b1+1
    c1=c1+1
  }
}
```

ACT.3.-

```
|formula_general<-function(a=3,b=8,c=6)
|{
|  q=b^2-4*a*c
|  if(q>0)
|  {
|    w=sqrt(q)
|    e=-b+w
|    r=e/2*a
|    print(r)
|    t=-b-w
|    y=t/2*a
|    print(y)
|  }
|  else
|  {
|    print("no se puede sacar raiz de un numero negativo")
|  }
|}
|formula_general()
```

ACT.4.-

```
|vector_formula_general<-function(y){
|
|  y<-c(1,2,3,4,5,6,7,8,9)
|  for(i in 1:9)
|  {
|    if(i%%3==0){
|      a=y[i]
|      b=y[i-1]
|      c=y[i-2]
|
|      q=b^2-4*a*c
|      if(q>0)
|      {
|        w=sqrt(q)
|        e=-b+w
|        r=e/2*a
|        print(r)
|        t=-b-w
|        y=t/2*a
|        print(y)
|      }
|      else
|      {
|        print("no se puede sacar raiz de un numero negativo")
|      }
|    }
|  }
|}
|vector_formula_general()
```

ACT.5.-

```
par<-function()  
{  
  for (i in 1:100){  
    if(i%%2==0){  
      print(i)  
    }  
    else  
    {  
      print("es impar")  
    }  
  }  
}  
par()
```

ACT.6.-

```
impar<-function()  
{  
  for (i in 1:100){  
    if(i%%2!=0){  
      print(i)  
    }  
    else  
    {  
      print("es par")  
    }  
  }  
}  
impar()
```

ACT.7.-

```
par1<-function(a=53)  
{  
  for (i in 1:a){  
    if(i%%2==0){  
      print(i)  
    }  
    else  
    {  
      print("es impar")  
    }  
  }  
}  
par1()
```

ACT.8.-

```
impar1<-function(a=53)
{
  for (i in 1:a){
    if(i%%2!=0){
      print(i)
    }
    else
    {
      print("el numero espar")
    }
  }
}
impar1()
```

ACT.9.-

```
act9<- function(n=45)
{
  x=3*n+1
  print(x)
}
act9()
```

ACT.10.-

```
act10<- function()
{
  x<-c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)
  for(i in 1:20){
    if(i%%3==0){
      x1=x[i]
      h=x1-1
      y=x1+h
      print(y)
    }
  }
}
act10()
```

ACT.11.-

```
x1<- c(1,3,5,7,9)
x2<- c(2,4,6,8,1)
x3<-c(4,5,6,7,8)
x4<-c(3,2,4,3,5)
x5<-c(6,7,5,9,3)
filas_juntas<-rbind(x1,x2,x3,x4,x5)
filas_juntas

M1<-matrix(filas_juntas,nrow = 5,ncol = 5,byrow = TRUE)
a<- array(c(M1),dim = c(5,5,2))
suma_de_arreglo<-function(x=0){
  for (i in 1:5) {
    for (j in 1:5) {
      if(i==j){
        x=x+M1[i,j]
      }
    }
  }
  print(x)
}
suma_de_arreglo()
```

ACT.12.-

```
numeros_primos<- function(num)
{
  current<-num
  ret.vals<-vector()
  x<- 2
  while(x<=num-1){
    while (current%%x==0) {
      current<-current/x
      ret.vals<-c(ret.vals,x)
    }
    x<-x+1
  }
  if(is.logical(ret.vals))return(num) else return(ret.vals)
}
numeros_primos(12)
```

ATC.14.-

```
esprimo <- function(n) {  
  if (n > .Machine$integer.max) {  
    cat("El número excede la capacidad\n")  
    return(NULL)  
  }  
  
  if (n==1 || n==2) return(T)  
  
  return(all(n %% (2:sqrt(n)) != 0))  
}  
esprimo()
```

ACT.15.-

```
fibonacci<- function(N)  
{  
  n<- 1  
  N<- as.integer(N)  
  if(N<1){  
    stop("proporcione valores positivos para 'n'")  
  }  
  f1<- -1  
  f2<- 1  
  while(n<=N){  
    fn<-f1+f2  
    n<-n+1  
    print(fn)  
    f1<- f2  
    f2<- fn  
  }  
}  
fibonacci(5)
```