

```

prueba (real m, real n)
real x,y,z
{retorno= 0
z= m+n
x= z /2
y=n
If (m<>0) and (n<>0) then z= m +1
else
if (z>0) then retorno= 0
i=1
while (i<=m)
i=m+1
retorno=i+z
Return(retorno)}

```

2.3. Defina los casos de prueba para probar todas las sentencias:

Probando los caminos que hagan pasar al menos 1 vez por cada línea de código, CP_A y CP_C permiten cumplir con el criterio de prueba

```
prueba (real m, real n)
```

```
real x,y,z
```

```
{retorno= 0
```

```
z= m+n
```

```
x= z /2
```

```
y=n
```

```
If (m<>0) and (n<>0) then
```

```
z= m +1
```

```
else
```

```
if (z>0) then
```

```
retorno= 0
```

```
i=1
```

```
while (i<=m)
```

```
i=m+1
```

```
retorno=i+z
```

```
Return(retorno)}
```

m	n
4	0
4	-2

m	n	x	y	z	i	m<>0	an	n<>0	z>0	i<=m
							d			

2.4 Defina los casos de prueba para probar todas las decisiones

DECISIÓN: $(m < > 0)$ and $(n < > 0)$ prueba (real m, real n)

m	n	DECISIÓN
		V
		F

DECISIÓN: $(z > 0)$

m	n	DECISIÓN
		F
		V

DECISIÓN: while $(i \leq m)$

m	n	DECISIÓN
		V
		F

real x,y,z
{retorno= 0
z= m+n
x= z /2

y=n

If $(m < > 0)$ and $(n < > 0)$ then
z= m +1

else

if $(z > 0)$ then
retorno= 0

i=1

while $(i \leq m)$
i=m+1

retorno=i+z
Return(retorno)}

m	n	x	y	z	i	m<>0	an	n<>0	z>0	i<=m
							d			

Defina los casos de prueba para probar todas las condiciones

Condición: $(m < > 0)$

m	n	CONDICIÓN
		V
		F

Condición: $(n < > 0)$

m	n	CONDICIÓN
		V
		F

Condición: $(z > 0)$

m	n	CONDICIÓN
		V
		F

Condición: $(i \leq m)$

m	n	CONDICIÓN
		V
		F

prueba (real m, real n)
real x,y,z
{retorno= 0

z= m+n

x= z /2

y=n

If $(m < > 0)$ and $(n < > 0)$ then
z= m +1

else

if $(z > 0)$ then
retorno= 0

i=1

while $(i \leq m)$
i=m+1

retorno=i+z
Return(retorno)}

m	n	x	y	z	i	m<>0	an	n<>0	z>0	i<=m
							d			

Defina los casos de prueba para probar todas las DECISIONES Y condiciones

Condición: $(m < > 0)$

m	n	CONDICIÓN
		V
		F

prueba (real m, real n)

real x,y,z

{retorno= 0

z= m+n

x= z /2

y=n

If $(m < > 0)$ and $(n < > 0)$ then

z= m +1

else

if $(z > 0)$ then

retorno= 0

i=1

while $(i \leq m)$

i=m+1

retorno=i+z

Return(retorno)}

Condición: $(n < > 0)$

m	n	CONDICIÓN
		V
		F

DECISIÓN: $(m < > 0)$ and $(n < > 0)$

m	n	DECISIÓN
		V
		F

Condición: $(z > 0)$

m	n	CONDICIÓN
		V
		F

Condición: $(i \leq m)$

m	n	CONDICIÓN
		V
		F

m	n	x	y	z	i	$m < > 0$	and	$n < > 0$	$z > 0$	$i \leq m$
0	0					f		f	f	

1	1					v		V	v	
---	---	--	--	--	--	---	--	---	---	--

2.6 Defina los casos de prueba para probar las condiciones múltiples

Condición múltiple: : $(m < > 0)$ and $(n < > 0)$

prueba (real m, real n)

m	n	$m < > 0$	and	$n < > 0$
		V	v	V
		V	f	F
		F	f	V
		F	f	F

real x,y,z

{retorno= 0

z= m+n

x= z /2

y=n

If $(m < > 0)$ and $(n < > 0)$ then

z= m +1

else

if $(z > 0)$ then

retorno= 0

i=1

while $(i \leq m)$

i=m+1

retorno=i+z

Return(retorno)}

m	n	x	y	z	i	$m < > 0$	and	$n < > 0$	$z > 0$	$i \leq m$
								d		

```

prueba (real m, real n)
real x,y,z
{retorno= 0 ①
z= m+n ②
x= z /2 ③
y=n ④
If (m<>0) and (n<>0) ⑤ then z= m +1 ⑥
else ⑦
if (z>0) then ⑧ retorno= 0 ⑨
i=1 ⑩
while (i<=m) ⑪
i=m+1 ⑫
retorno=i+z ⑬
Return(retorno)} ⑭ ⑮

```

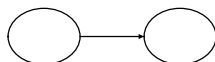
```

prueba (real m, real n)
real x,y,z
{retorno= 0
z= m+n
x= z /2
y=n
If (m<>0) and (n<>0) then
    z= m +1
else
    if (z>0) then
        retorno= 0
i=1
while (i<=m)
    i=m+1
    retorno=i+z
Return(retorno)
}

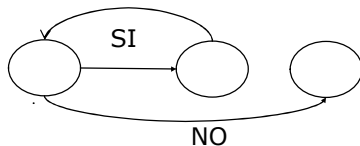
```

Notación

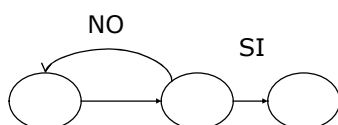
Secuencia



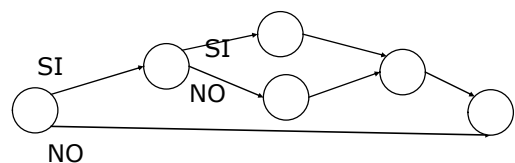
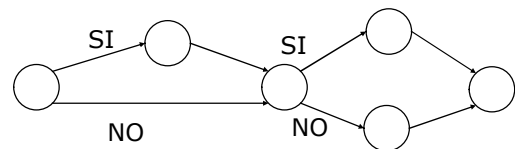
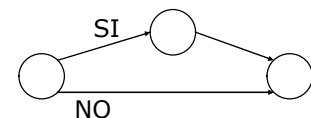
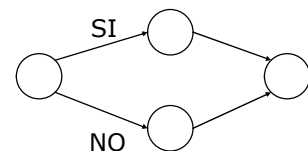
Bucle While/for



Bucle Hasta



Condición IF



Aristas – nodos +2=

```
prueba (real m, real n)
real x,y,z
{retorno= 0
z= m+n
x= z /2
y=n
If (m<>0) and (n<>0) then
z= m +1
else
if (z>0) then retorno= 0
i=1
while (i<=m)
i=m+1
retorno=i+z
Return(retorno)}
```

Aristas: relaciones entre nodos.
Nodos: bloque de 1 a n líneas de código

Aristas – nodos +2=

```
prueba (real m, real n)
real x,y,z
{retorno= 0
z= m+n
x= z /2
y=n
If (m<>0) and (n<>0) then
z= m +1
else
if (z>0) then
retorno= 0
i=1
while (i<=m)
i=m+1
retorno=i+z
Return(retorno)}
```

CP_A:
CP_B:
CP_C:
CP_D:

m	n	x	y	z	i	m<>0	and n<>0	z>0	i<=m	caminos
0	0	0	0	0	1	F	F	F	F	
4	0	2	0	4	4	V	F	F	V	V
					5					
4	-2	2	-2	-2	4	V	V	V	--	V
					5					
-2	-2	-1	-2	-4	1	V	V	v	--	F
					-1					

Aristas – nodos +2=

```
INT FUNCTION DIVISOR (A,B)
X,Y Z: real
{
If a<0 then
x= abs(a)
else x=a
If b<0 then
y= abs (b)
Else y=b
If (x-PARTENTERA(x)>0) OR (y-
PARTENTERA(y)>0) then retorno=0
Else
while x<> y do
{If x>y then
x=x-y
else x=y-x}
return (retorno)
}
```