



Programación II. I.T.I. de Gestión
Ejercicios de verificación de programas

Curso 2009/10

Ejercicio 1

Determine las precondiciones mas debiles que hacen parcialmente correctas las siguientes asignaciones.

1. $\{ \} \quad n := m \quad \{n \leq 100\}$
2. $\{ \} \quad n := 2 \times n \quad \{n \leq 100\}$
3. $\{ \} \quad n := n + 1 \quad \{n = n + 1\}$
4. $\{ \} \quad b := b \vee (k = 0) \quad \{-b\}$

Ejercicio 2

Determina las postcondiciones mas fuertes que puede garantizar para los siguientes fragmentos de programas.

1. $\{n > 0\} \wedge \{z + n \times y = x \times y\} \quad z := z + y; n := n - 1 \quad \{ \}$
2. $\{r + q \times y = x\} \wedge \{r - y \geq 0\} \quad r := r - y; q := q + 1 \quad \{ \}$

Ejercicio 3

Verifique la corrección total del siguiente algoritmo:

```
algoritmo maximo  $\{a, b, c \in \mathbb{R}\}$   
   $x := a$   
  si  $x < b$   
    entonces  $x := b$   
  fin_si  
  si  $x < c$   
    entonces  $x := c$   
  fin_si
```

Ejercicio 4

Verifique la corrección total del procedimiento *ordenar*:

```
Procedure intercambiar (var  $x, y$ : integer);  
  var  
     $w$ : integer;  
  begin
```



```
w:=x; x:=y; y:=w  
end;
```

```
Procedure Ordenar(var x,y,z: integer);  
begin  
  if (x>y) then intercambiar(x,y);  
  if (x>z) then intercambiar(x,z);  
  if (y>z) then intercambiar(y,z);  
end;
```

Ejercicio 5

Verifique la corrección total de los siguientes procedimientos:

```
Procedure A(x,y: integer);  
var  
  z,r: integer;  
begin  
  r:= 0; z:= x;  
  while (r<>y) do  
    begin  
      z:= z+1;  
      r:= r+1  
    end  
  end  
end;
```

```
Procedure B(x,y: integer);  
var  
  z,r: integer;  
begin  
  r:= 0; z:= x;  
  while (z<y) do  
    begin  
      z:= z+1;  
      r:= r+1  
    end  
  end  
end;
```

Ejercicio 6

Verifique la corrección total de la siguiente función teniendo en cuenta las siguientes propiedades del máximo común divisor:

- $\forall u, v \in \mathbb{N} \text{ mcd}(u, u) = u$
- $\forall u, v \in \mathbb{N} \text{ mcd}(u, v) = \text{mcd}(v, u)$
- $\forall u, v \in \mathbb{N} \mid u > v \text{ mcd}(u, v) = \text{mcd}(u - v, v)$

```
Function MCD(x,y: integer): integer;  
var  
  a,b: integer; {x > 0, y > 0}  
begin
```



```
a:= x; b:= y;
while (a<>b) do
  if a>b then a:= a-b else b:= b-a;
MCD:= a {a = b = MCD(x,y)}
end;
```

Ejercicio 7

Verifique la corrección total de la siguiente función:

```
Function Expo (x, eps: real): real;
{1 > x > 0, eps > 0}
var
  t, s: real;
  n: integer;
begin
  t:= 1
  s:= t
  n:= 0
  while (t>eps) do
    begin
      {(s = 1 + x + ... + x^n/n!) ^ (t = x^n/n!)}
      n:= n+1;
      t:= t*x/n;
      s:= s+t
    end;
  Expo:= s
end;
```

Ejercicio 8

Dada la siguiente función, verificar la corrección parcial de la misma y establecer la condición inicial que debe cumplir u para que se verifique la corrección total:

```
Function Fact (u: integer): integer;
var
  f, k: integer;
begin {?}
  f:= 1;
  k:= 0;
  while (k <> u) do
    begin
      k:= k+1;
      f:= f*k
    end;
  Fact:= f {?}
end;
```

Ejercicio 9

Verifique la corrección total del siguiente algoritmo:



```
algoritmo examen(x, y : enteros) {x=x0, y=y0 > 1, x0, y0 ∈ ℕ}
inicio
  suma := 0
  repetir
    si odd(y) entonces suma := suma + x fin_si
    x := x × 2
    y := y div 2
  hasta_que y=1 {xy+suma=x0y0}
  suma := suma + x
fin
```

Ejercicio 10

Verificar la corrección parcial y total de la siguiente función considerando que un buen candidato a invariante puede ser $x \times y + z = x_0 \times y_0$

```
Function Prod(x, y: integer): integer;
var
  z: integer;
begin {x > 0}
  z := 0;
  while (x <> 0) do
    begin
      if (odd(x)) then z := z+y;
      x := x div 2;
      y := 2*y;
    end;
  Prod := z {z = xy}
end;
```

Ejercicio 11

Verificar la corrección parcial de los siguientes fragmentos de programa:

```
{x, y ∈ ℝ}
si y < x
  entonces z := x; x := y; y := z
fin_si
```

```
{x0, y0 ∈ ℤ+x}
y := y0 ; z := x0
mientras y ≠ 0 hacer
  y := y-1; z := z+1
fin_mientras
```

¿Qué se puede decir sobre la corrección total de los fragmentos de programa anteriores? Razonar la respuesta.

Ejercicio 12

Verificar la corrección parcial y total del siguiente procedimiento indicando qué hace:

```
procedure xx(x, y: integer); {x = x0 > 0 , y = y0 > 0}
var p, t: integer;
begin
  p := 0;
  repeat
```



```
t:=0;  
repeat  
  p:=p+1;  
  t:=t+1; {p = x0y0 - xy0 + t ; t ≤ y}  
until t=y;  
  x:=x-1; {p = x0y0 - xy0 ; x ≥ 0}  
until x=0  
end;
```

Ejercicio 13

Demostrar la corrección total del algoritmo reflejado en este diagrama de flujo con respecto a la precondition

$$\{n \in \mathbb{Z}, n > 0\}$$

y la postcondición

$$s = \sum_{i=1}^n i^2$$

Considere que todas las variables utilizadas en el algoritmo son de tipo entero, y que un buen candidato a invariante podría ser

$$\{(s = \sum_{i=1}^t i^2) \wedge (t \leq n)\}$$

Nota: Ejercicio de examen del curso 2008/2009

