3. Gaia: Programen Egiaztapena

4. Ariketa-orria: Iterazioen egiaztapena

1. Hurrengo baieztapenetan aukeratu zuzena den inbariantea:

1.1. Programa honek x elementua $A(1..n)$ bektorean agertzen den ala ez erabakitzen du.

$\{ n \geq 1 \}$
```
i := 0; dago := false;
```
INB1 $\equiv \{ (\mathit{dago} \leftrightarrow \exists j \; (1 \leq j \leq i \wedge A(j) = x)) \wedge 0 \leq i \leq n \}$ [ ]

INB2 $\equiv \{ (\mathit{dago} \wedge \exists j \; (1 \leq j \leq i \wedge A(j) = x)) \wedge 0 \leq i \leq n \}$ [ ]
```
while not dago and i < n loop
   i := i+1;
   if A(i) = x then
      dago := true;
   end if;
end loop;
```
$\{ \mathit{dago} \leftrightarrow \exists j \; (1 \leq j \leq n \wedge A(j) = x) \}$

1.2. Programa honek *lehen* aldagai boolearrean x zenbaki arrunta lehena den ala ez erabakitzen du.

$\{ x \geq 2 \}$
```
d := 2;
```
INB1 $\equiv \{ 1 < d \leq x \wedge \forall i \; (1 < i < d \rightarrow x \, \mathit{mod} \, i \neq 0) \}$ [ ]

INB2 $\equiv \{ 1 < d \leq x \leftrightarrow \forall i \; (1 < i < d \rightarrow x \, \mathit{mod} \, i \neq 0) \}$ [ ]
```
while x mod d /= 0 loop
   d := d+1;
end loop;
lehen := (d = x);
```
$\{ \mathit{lehen} \leftrightarrow \forall i \; (1 < i < x \rightarrow x \, \mathit{mod} \, i \neq 0) \}$

1.3. Programa honek x zenbaki arruntaren faktoriala kalkulatzen du.

$\{ x \geq 0 \}$

```
f := 1; t := x;
```

INB1 $\equiv \{ f = \prod_{i=1}^{t-1} i \ \wedge \ t \geq 0 \}$ [ ]

INB2 $\equiv \{ f = \prod_{i=t+1}^{x} i \ \wedge \ t \geq 0 \}$ [ ]

INB3 $\equiv \{ f = \prod_{i=t-1}^{x} i \ \wedge \ t \geq 0 \}$ [ ]

INB4 $\equiv \{ f = \prod_{i=t}^{x} i \ \wedge \ t \geq 0 \}$ [ ]

```
while t >= 1 loop
   f := f*t;
   t := t-1;
end loop;
```

$\{ f = \prod_{i=1}^{x} i \}$

2. Hurrengo iterazioen inbariantea eta borne adiezpena asmatu:

2.1. Honako programa honek $A(1..n)$ bektoreko minimoa kalkulatzen du m aldagaian.

$\{ n \geq 1 \}$

```
m := A(1); k := 1;
while k < n loop
   k := k+1;
   if A(k) < m then
      m := A(k);
   end if;
end loop;
```

INB $\equiv$ { ________________________ }

E $\equiv$ __________

$\{ txikiena(A(1..n), m) \}$

non:

$$txikiena(A(1..n), m) \equiv \exists i \ (1 \leq i \leq n \ \wedge \ A(i) = m \ ) \ \wedge \ \forall j \ (1 \leq j \leq n \ \rightarrow \ A(j) \geq m \ )$$

2.2. Programa honek $A(1..n)$ taulako elementuak atzekoz aurrera jartzen ditu.

$\{\ n \geq\ 1\ \wedge\ A = (a_1, \ldots, a_n)\ \}$

```
k := 1;
```

INB $\equiv \{$ ____ $\}$

E $\equiv$ ___________

```
while k <= n/2 loop
   lag := A(k);
   A(k) := A(n-k+1);
   A(n-k+1) := lag;
   k := k+1;
end loop;
```

$\{\ \forall i\ (\ 1 \leq i \leq n\ \rightarrow\ A(i) = a_{n-i+1}\ )\ \}$

2.3. A taulako elementuen erdiak baino gehiago, B taulan posizio berean daudenak baino handiagoak diren ala ez adieraziko du b aldagai boolearrak.

$\{\ n \geq 1\ \}$

```
i := 1; z := 0;
```

INB $\equiv \{$ ____ $\}$

E $\equiv$ ___________

```
while i <= n loop
   if A(i) > B(i) then
      z := z+1;
   end if;
   i := i+1;
end loop;
b := (z > n/2);
```

$\{\ b\ \leftrightarrow\ \mathcal{N}j\ (\ 1 \leq j \leq n\ \wedge\ A(j) > B(j)\ ) > \frac{n}{2}\ \}$

3. Dokumentatu markatzen diren asertzioekin honako programa iteratibo hauek:

3.1. Honako programa honek $|x - y|$ adierazpenaren balioa uzten du d aldagaian.

```
Aurre ≡ { ________________________________________ }
   d := 0;
   if x <= y then
      u := x;
      z := y;
   else
      u := y;
      z := x;
   end if;
   φ1 ≡ { ________________________________________ }
         INB ≡ { ____________________________________ }
         E ≡ ____________
   while u /= z loop
      φ2 ≡ { ________________________________________ }
      z := z-1;
      φ3 ≡ { ________________________________________ }
      d := d+1;
   end loop;
Post ≡ { d = |x − y| }
```

3.2. Programak $A(1..n)$ bektorean bakoitiak diren osagaien kopurua eta bikoitiak direnena berdina den ala ez erabakitzen du.

```
Aurre ≡ { ________________________________________ }
   i := 0; w := 0; z := 0;
      INB ≡ { ________________________________________ }
   while ( i < n ) loop          E ≡ ____________
      i := i+1;
         φ1 ≡ { ________________________________________ }
      if ( A(i) mod 2 = 0 ) then
         φ2 ≡ { ________________________________________ }
         w := w+1;
      else
         φ3 ≡ { ________________________________________ }
         v := v+1;
      end if;
   end loop;
      φ4 { ________________________________________ }
   e := (w=v);
Post ≡ { ________________________________________ }
```

3.3. Programak 2ren 0 eta n-ren arteko berreduren batura kalkulatzen du b aldagaian; alegia, $2^0, 2^1, \ldots, 2^n$ segidaren batura kalkulatzen du.

Aurre $\equiv$ { ________________________________ }
```
   i := 0; p := 1; b := 1;
   while i < n loop     INB ≡ { ____________________ }
                        E ≡ { ________ }
         φ1 ≡ { ________________________________ }
      i := i + 1;
         φ2 ≡ { ________________________________ }
      p := p * 2;
         φ3 ≡ { ________________________________ }
      b := b + p;
   end loop;
```
Aurre $\equiv$ { ________________________________ }

3.4. $A(1..n)$ taula batura berdineko bi sekziotan bana daitekeen ala ez erabakiko du programa honek. Hala bada, i aldagaia izango da banaketa adierazten duen indizea.

Aurre $\equiv$ { ________________________________ }
```
   x := A(1); y := 0; k:= 2;
         INB ≡ { ____________________________ }
         E ≡ { ________ }
   while k <= n loop
      y := y+A(k);
      k := k+1;
   end loop;
   i := 1;
         INB ≡ { ____________________________ }
         E ≡ { ________ }
   while x /= y and i < n loop
      i := i+1;
      x := x+A(i);
      y := y-A(i);
   end loop;
   eqsum := (not i = n);
```
Post $\equiv$ { ________________________________ }

4. Programa honek batura eta biderkadura berdina duen $A(1..j)$ sekziorik luzeena mugatzen duen j indizea k aldagaian uzten du. Zuzentasun osoaren frogapenaren eskema asmatu.

```
m := 1; k := 1; bat := A(1); bider := A(1);
while m < n loop
   m := m+1; bat := bat+A(m); bider := bider*A(m);
   if bat = bider then
      k := m;
   end if;
end loop;
```

5. Hurrengo programako iterazioak inbariantea konserbatzen duela frogatu. Programak x elementua $A(1..n)$ bektorean agertzen den ala ez erabakitzen du.

$\{ \; n \geq 1 \; \}$

```
i := 0; dago := false;
while not dago and i < n loop
   i := i+1;
   if A(i) = x then
      dago := true;
   end if;
end loop;
```

$\{ \; dago \leftrightarrow \exists j \; (\; 1 \leq j \leq n \; \wedge \; A(j) = x \;) \; \}$

6. Hurrengo programen zuzentasun osoa frogatu.

6.1. Honako programa honek x eta y zenbaki osokoen biderkadura kalkulatzen du.

```
z := 0;
while x /= 0 loop
   z := z+y;
   x := x-1;
end loop;
```

6.2. Programa honek x elementua $A(1..n)$ bektorean zenbat aldiz agertzen den kontatzen du.

```
i := 1; z := 0;
while i <= n loop
   if A(i) = x then
      z := z+1;
   end if;
   i := i+1;
end loop;
```