3. Gaia: Programen egiaztapena

3. Ariketa-orria: Asignazioak, konposaketa sekuentziala eta baldintzazkoak

1. Ondoko baieztapenetan post-baldintzak ({ ___ }) bete:

1.1. $\{ \; true \; \}$

```
if x > y then
   z := x;
else
   z := y;
end if;
```

$\{$ ______________________________ $\}$

1.2. $\{ \; 1 \leq i \leq n \; \wedge \; z = \mathcal{N}k \; (\; 1 \leq k < i \; \wedge \; A(k) > B(k) \;) \; \}$

```
if A(i) > B(i) then
   z := z+1;
end if;
```

$\{$ ______________________________ $\}$

2. Idatzi post-baldintza bete dadin exekutatu beharreko agindua (____;):

2.1. $\{ \; i = n \; \wedge \; x \notin A(1..n-1) \; \}$

```
if ____________ then
   ______________________;
else
   ______________________;
end if;
```

$\{ \; dago \leftrightarrow x \in A(1..n) \; \}$

2.2. $\{ \; 1 \leq i < n \; \wedge \; m = max(A(1..i)) \; \}$

```
i := i+1;
if ____________ then
   ______________________;
end if;
```

$\{ \; 1 \leq i \leq n \; \wedge \; m = max(A(1..i)) \; \}$

3. Hurrengo frogapenetan hutsuneak bete (___):

3.1. $\{ \; 1 \leq i < n \; \wedge \; \neg dago \; \wedge \; x \notin A(1..i) \; \}$

```
i := i+1;
if A(i) = x then
   dago := true;
end if;
```

$\{ \; 1 \leq i \leq n \; \wedge \; (\; dago \leftrightarrow x \in A(1..i) \;) \; \}$

Frogapena:

1. $(\ 1 \le i < n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i)\ )$
$\rightarrow\ (\ 1 \le i+1-1 < n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i+1-1)\ )$

2. $\{\ 1 \le i+1-1 < n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i+1-1)\ \}$
```
i := i+1;
```
$\{$ ________________________________ $\}$ **(AA)**

3. $(\ 1 \le i-1 < n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i-1)\ )$
$\rightarrow\ (\ 1 \le i \le n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i-1)\ )$

4. $\{\ 1 \le i < n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i)\ \}$
```
i := i+1;
```
$\{$ ________________________________ $\}$ 1, 2, 3 eta **(ODE)**

5. $(\ 1 \le i \le n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i-1) \ \wedge\ A(i) = x\ )$
$\rightarrow\ (\ 1 \le i \le n \ \wedge\ x \in A(1..i)\ )$
$\rightarrow\ (\ 1 \le i \le n \ \wedge\ (\ true \leftrightarrow x \in A(1..i)\ )\ )$

6. $\{$ ________________________________ $\}$
```
dago := true;
```
$\{$ ________________________________ $\}$ **(AA)**

7. $\{\ 1 \le i \le n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i-1) \ \wedge\ A(i) = x\ \}$
```
dago := true;
```
$\{\ 1 \le i \le n \ \wedge\ (\ dago \leftrightarrow x \in A(1..i)\ )\ \}$ 5, 6 eta **(ODE)**

8. $($ ____ $)$
$\rightarrow\ (\ 1 \le i \le n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i)\ )$
$\rightarrow\ (\ 1 \le i \le n \ \wedge\ (\ dago \leftrightarrow x \in A(1..i)\ )\ )$

9. $(\ 1 \le i \le n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i-1)\ ) \ \rightarrow\ ($ __________ $)$
$\rightarrow\ def(A(i) = x)$

10. $\{\ 1 \le i \le n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i-1)\ \}$
```
if A(i) = x then
   dago := true;
end if;
```
$\{\ 1 \le i \le n \ \wedge\ (\ dago \leftrightarrow x \in A(1..i)\ )\ \}$ ________________

11. $\{\ 1 \le i < n \ \wedge\ \neg dago \ \wedge\ x \notin A(1..i)\ \}$
```
i := i+1;
if A(i) = x then
   dago := true;
end if;
```
$\{\ 1 \le i \le n \ \wedge\ (\ dago \leftrightarrow x \in A(1..i)\ )\ \}$ 4, 10 eta **(KPE)**

3.2. $\{ 1 \leq err < n \ \wedge \ 1 \leq zut \leq n \ \wedge \ (err-1) \times n + zut = b \}$

```
if zut = n then
   err := err+1;
   zut := 1;
else
   zut := zut+1;
end if;
```

$\{ 1 \leq err, zut \leq n \ \wedge \ (err-1) \times n + zut = b+1 \}$

Frogapena:

1. $(1 \leq err < n \ \wedge \ 1 \leq zut \leq n \ \wedge \ (err-1) \times n + zut = b \ \wedge \ zut = n)$
   $\rightarrow (1 \leq err < n \ \wedge \ (err-1) \times n + n = b)$
   $\rightarrow (1 \leq err < n \ \wedge \ err \times n = b)$
   $\rightarrow (1 \leq err + 1 - 1 < n \ \wedge \ (err + 1 - 1) \times n = b)$
2. { ______________________________ }
   `err := err+1;`
   $\{ 1 \leq err - 1 < n \ \wedge \ (err-1) \times n = b \}$ **(AA)**
3. $(1 \leq err - 1 < n \ \wedge \ (err-1) \times n = b)$
   $\rightarrow (1 \leq err \leq n \ \wedge \ (err-1) \times n = b)$
4. $\{ 1 \leq err < n \ \wedge \ 1 \leq zut \leq n \ \wedge \ (err-1) \times n + zut = b \ \wedge \ zut = n \}$
   `err := err+1;`
   $\{ 1 \leq err \leq n \ \wedge \ (err-1) \times n = b \}$ 1, 2, 3 eta **(ODE)**
5. $\{ 1 \leq err \leq n \ \wedge \ (err-1) \times n = b \}$
   `zut := 1;`
   { ______________________________ } **(AA)**
6. $(1 \leq err \leq n \ \wedge \ (err-1) \times n = b \ \wedge \ zut = 1)$
   $\rightarrow (1 \leq err \leq n \ \wedge \ (err-1) \times n + 1 = b + 1 \ \wedge \ zut = 1)$
   $\rightarrow (1 \leq err, zut \leq n \ \wedge \ (err-1) \times n + zut = b + 1)$
7. $\{ 1 \leq err \leq n \ \wedge \ (err-1) \times n = b \}$
   `zut := 1;`
   $\{ 1 \leq err, zut \leq n \ \wedge \ (err-1) \times n + zut = b+1 \}$ 5, 6 eta **(ODE)**
8. $\{ 1 \leq err < n \ \wedge \ 1 \leq zut \leq n \ \wedge \ (err-1) \times n + zut = b \ \wedge \ zut = n \}$
   `err := err+1;`
   `zut := 1;`
   $\{ 1 \leq err, zut \leq n \ \wedge \ (err-1) \times n + zut = b+1 \}$ __________
9. (____)
   $\rightarrow (1 \leq err < n \ \wedge \ 1 \leq zut < n \ \wedge \ (err-1) \times n + zut = b)$
   $\rightarrow (1 \leq err < n \ \wedge \ 1 \leq zut + 1 - 1 < n \ \wedge$
   $(err-1) \times n + zut + 1 - 1 = b)$
10. { ____ }
   `zut := zut+1;`
   $\{ 1 \leq err < n \ \wedge \ 1 \leq zut - 1 < n \ \wedge \ (err-1) \times n + zut - 1 = b \}$ **(AA)**

11. $(1 \leq err < n \ \wedge \ 1 \leq zut-1 < n \ \wedge \ (err-1) \times n + zut - 1 = b \ )$
$\rightarrow \ (1 \leq err < n \ \wedge \ 1 \leq zut \leq n \ \wedge \ (err-1) \times n + zut = b+1 \ )$
$\rightarrow \ (1 \leq err, zut \leq n \ \wedge \ (err-1) \times n + zut = b+1 \ )$

12. $\{ \ 1 \leq err < n \ \wedge \ 1 \leq zut \leq n \ \wedge \ (err-1) \times n + zut = b \ \wedge \ zut \neq n \ \}$

```
zut := zut+1;
```

$\{ \ 1 \leq err, zut \leq n \ \wedge \ (err-1) \times n + zut = b+1 \ \}$ — 9, 10, 11 eta **(ODE)**

13. $(1 \leq err < n \ \wedge \ 1 \leq zut \leq n \ \wedge \ (err-1) \times n + zut = b \ ) \ \rightarrow$
$def(zut = n)$

14. $\{ \ 1 \leq err < n \ \wedge \ 1 \leq zut \leq n \ \wedge \ (err-1) \times n + zut = b \ \}$

```
if zut = n then
   err := err+1;
   zut := 1;
else
   zut := zut+1;
end if;
```

$\{ \ 1 \leq err, zut \leq n \ \wedge \ (err-1) \times n + zut = b+1 \ \}$ — 8, 12, 13 eta **(BDE)**

4. Egiaztatu ondoko baieztapenak:

4.1. $\{ \ 1 < i \leq n \ \wedge \ m = max(A(1..i-1)) \ \}$

```
if m < A(i) then
   m := A(i);
end if;
i := i+1;
```

$\{ \ 1 < i \leq n+1 \ \wedge \ m = max(A(1..i-1)) \ \}$

4.2. $\{ \ true \ \}$

```
if x mod 2 = 0 then
   bikoiti := true;
else
   bikoiti := false;
end if;
```

$\{ \ bikoiti \leftrightarrow x \ mod \ 2 = 0 \ \}$

5. Frogatu zuzentasunari buruzko ondorengo baieztapenak, kontuan hartuz FIB_i-k Fibonacciren segidaren i-garren terminoa adierazten duela:

5.1. $\{ \ 1 \leq i < n \ \wedge \ x = FIB_i \ \wedge \ y = FIB_{i+1} \ \wedge \ z = FIB_{i+2} \ \}$

```
x := y;
y := z;
z := x+y;
i := i+1;
```

$\{ \ 1 \leq i \leq n \ \wedge \ x = FIB_i \ \wedge \ y = FIB_{i+1} \ \wedge \ z = FIB_{i+2} \ \}$

5.2. $\{\ \exists i\ (\ i \geq 0\ \wedge\ u = FIB_i\ \wedge\ z = FIB_{i+1}\ )\ \}$
```
  u := u+z;
  z := u+z;
```
$\{\ \exists i\ (\ i \geq 0\ \wedge\ u = FIB_i\ \wedge\ z = FIB_{i+1}\ )\ \}$

6. Izan bedi ondoko Hoare-ren hirukotea:

$\{\ true\ \}$
```
  if x = 0 then
    y := 3;
  end if;
```
$\{\ \varphi\ \}$

Ondokoetatik zein da post-baldintza egokia?

$\{\ x = 0\ \wedge\ y = 3\ \}$

$\{\ x = 0\ \leftrightarrow\ y = 3\ \}$

$\{\ x = 0\ \rightarrow\ y = 3\ \}$

$\{\ y = 3\ \rightarrow\ x = 0\ \}$

7. Zuzena al da ondoko baieztapena?

$\{\ true\ \}$
```
  if x < y then
    lag := x; x := y; y := lag;
  elsif y < z then
    lag := y; y := z; z := lag;
  else
    lag := x; x := z; z := lag;
  end if;
```
$\{\ x \geq y \geq z\ \}$

Eta beste hau?

$\{\ true\ \}$
```
  if x < y then
    lag := x; x := y; y := lag;
  end if;
  if y < z then
    lag := y; y := z; z := lag;
  end if;
  if x < y then
    lag := x; x := y; y := lag;
  end if;
```
$\{\ x \geq y \geq z\ \}$

8. Asmatu inferentzi erregela egokia honako aginduentzat:

8.1.

```
if B1 then I1;
elsif B2 then I2;
   ...
elsif Bn then In;
else In+1;
end if;
```

Kasu-hautaketa aginduen aldaera desberdinak dira honako hauek. E datu-mota diskretu bateko espresioa da, eta b_i balioak berekoak dira (bere artean desberdinak).

8.2.

```
case E is
   when b1 => I1;
      ...
   when bn => In;
end case;
```

P_i exekutatzen da E b_i denean. E ebaluatzean $b_1,\ldots,b_n$ ez den balioa lortzen bada, ez da ezer exekutatzen.

8.3.

```
case E is
   when b1 => I1;
      ...
   when bn => In;
   when others => In+1;
end case;
```

P_i exekutatzen da E b_i denean. E ebaluatzean $b_1,\ldots,b_n$ ez den balioa lortzen bada, P_{n+1} exekutatzen da.