Tipos Abstractos de Datos. Segunda relación de problemas. Tema 1 (03-04)

1.– Suponiendo las siguientes definiciones de variables Java

```
int i, j, k, n, m, s, t, u, v, w, x, y, z;
int a[], b[];
Boolean encontrado, capicua;
```

se pide verificar formalmente los siguientes programas:

a) $\{Q \equiv 0 \leq n < \texttt{a.length}\}$
```
i = 0;
while ((i<=n)&&(a[i]!=x)) i++;
encontrado = (i<=n);
```
$\{R \equiv \mathit{encontrado} = (\exists \alpha \in \{0..n\} \bullet ((a[\alpha] = x) \wedge (\forall \beta \in \{0..\alpha - 1\} \bullet a[\beta] \neq x)))\}$

b) $\{Q \equiv (m > 0) \wedge (n > 0)\}$
```
x = m;
y = n;
while (x!=y)
  if (x<y) y = y-x;
  else x = x-y;
```
$\{R \equiv x = mcd(m, n)\}$

c) $\{Q \equiv x \geq 0\}$
```
u = 0; v = 1; w = 1;
while (w<=x) {
  u++; v+=2; w+=v;
};
```
$\{R \equiv ((u * u) \leq x) \wedge (x < (u + 1) * (u + 1)\}$

d) $\{Q \equiv 1 \leq n \leq \texttt{a.length}\}$
```
i = 0;
encontrado = false;
while ((i<n)&&(!encontrado))
  if (a[i]==x) encontrado = true;
```
$\{R \equiv \mathit{encontrado} = (\exists \alpha \in \{0..n-1\} \bullet ((a[\alpha] = x) \wedge (\forall \beta \in \{0..\alpha - 1\} \bullet a[\beta] \neq x)))\}$

e) $\{Q \equiv (m \geq 0) \wedge (n \geq 0)\}$
```
x = m; y = n; z = 1;
while (y>0)
  if ((y%2)==0) { y = y/2; x = x*x; };
  else { y--; z = z*x; };
```
$\{R \equiv (z = m^n)\}$

f) $\{Q \equiv 1 \leq n \leq \texttt{a.length}\}$
```
for (s=a[0],i=1; i<n; i++) s+=a[i];
```
$\{R \equiv s = (\Sigma \alpha \in \{0..n-1\} \bullet (a[\alpha]))\}$

g) $\{Q \equiv 1 \leq n \leq \texttt{a.length}\}$

```
i = 1; m = a[0];
while (i<n) {
  if (a[i]<m) m = a[i];
  i++;
};
```

$\{R \equiv (\forall \alpha \in \{0..n-1\} \bullet (a[\alpha] \geq m)) \wedge (\exists \beta \in \{0..n-1\} \bullet a[\beta] = m))\}$

h) $\{Q \equiv n \geq 0\}$

```
int f(int n) {
  int res;
  if (n==0) res = 1;
  else res = n*f(n-1);
  return res;
};
```

$\{R \equiv res = n!\}$

i) $\{Q \equiv 1 \leq n \leq \texttt{a.length}\}$

```
i = 0; j = n-1; capicua = true;
while ((i<j)&&capicua) {
  capicua = (a[i]==a[j]);
  i++; j--;
};
```

$\{R \equiv capicua = (\forall \alpha \in \{0..(n/2-1)\} \bullet (a[\alpha] = a[n-\alpha-1]))\}$

j) $\{Q \equiv (n > 0) \wedge (m > 0)\}$

```
class Resultado { int q; int r; };
Resultado f(int m, int n){
  Resultado res;
  if (m<n) {
      res = new Resultado();
      res.q=0;
      res.r=m;
  } else {
      res = f(m-n,n);
      res.q++;
  };
  return res;
};
```

$\{R \equiv (m = n * q + r) \wedge (0 \leq r < n)\}$

k) $\{Q \equiv n \geq 0\}$

```
int sumaDigitos(int n) {
  if (n<10) return n;
  return (n%10) + sumaDigitos(n/10);
};
```

$$\{R \equiv res = \sum_{i=0}^{\lfloor \log_{10} n \rfloor} (n \mathrm{DIV} 10^i) \mathrm{MOD} 10\}$$

2.– Dado un vector a de enteros con n elementos ($n \geq 1$), diseñar una función que calcule su máximo y verifíquese formalmente.

3.– Diseñar y verificar formalmente una función que calcule la moda de un vector de enteros, con n elementos ($n \geq 1$).

4.– *El rellano más largo (E.Gries).* Dado un vector a no decreciente de n enteros con $n \geq 1$, diseñar una función que calcule la longitud de su rellano más largo. Un rellano es un tramo de valores consecutivos iguales. Verificar formalmente dicha función.

5.– *La bandera holandesa (E.W.Dijkstra).* Se dispone de una hilera de n bolas azules, rojas y blancas, con $n \geq 0$. Se dispone también de un brazo mecánico para manipular las bolas que puede realizar, bajo control de un programa, las dos funciones siguientes:

a) `color(i)`, con $1 \leq i \leq n$, que devuelve el color de la bola i.

b) `permuta(i,j)`, con $1 \leq i \leq n$,$1 \leq j \leq n$, que intercambia de posición las bolas i y j.

Diseñar y verificar formalmente un algoritmo que coloque las bolas según su color: azul a la izquierda, blanco en el centro y rojo a la berecha.

6.– Dado un vector a de n enteros con $n \geq 1$, y dos enteros x e y, diseñar y verificar formalmente una función que sustituya en el vector a todas las apariciones de x (si las hubiese) por y.

7.– Basándose en los problemas de la relación anterior, diseñar y verificar formalmente una función que, dado un número natural n, decida si es un número "guay".

8.– Basándose también en los problemas de la relación anterior, dado un vector a de n enteros con $n \geq 1$, diseñar y verificar formalmente dos funciones que decidan si el vector a es "gaspariforme" o "melchoriforme".

9.– Diseñar una versión recursiva y otra iterativa de una función que calcule la parte entera del logaritmo en base b, $b > 1$, de un número natural n. Verificar formalmente ambas versiones.

10.– Diseñar y verificar formalmente una función recursiva lineal que calcule el n-ésimo número de la sucesión de Fibonacci.