Patterns of Algorithms – Examples 2

Example 1 – Copy

Absolute values: Copy the absolute values of elements from a given array into the new array.

Specification:

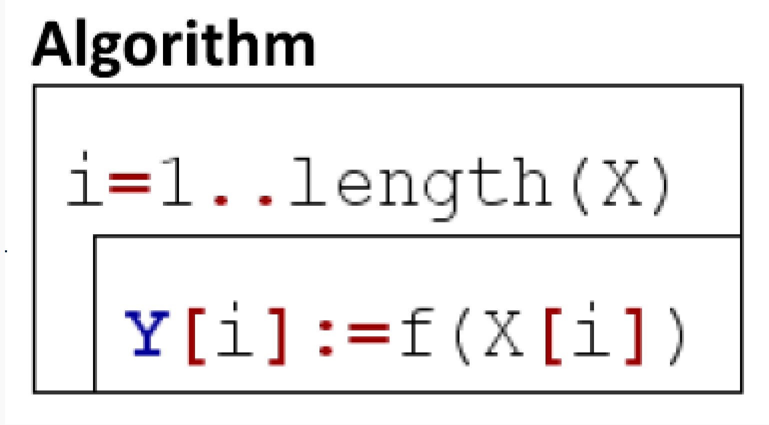
Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{Z}^N$

f: $\mathbb{Z} \rightarrow \mathbb{N}$

Output: $Y_{1..N} \in \mathbb{N}^N$

Precondition:

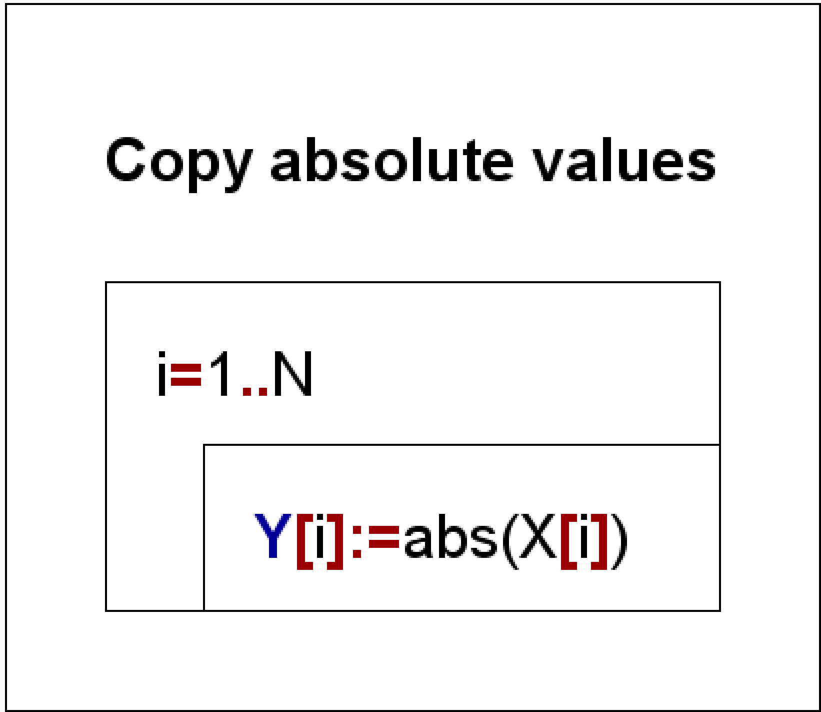
Postcondition: $\forall\, i\ (1 \leq i \leq N) : Y_i = ABS(X_i)$



Algorithm

```
i=1..length(X)
    Y[i]:=f(X[i])
```

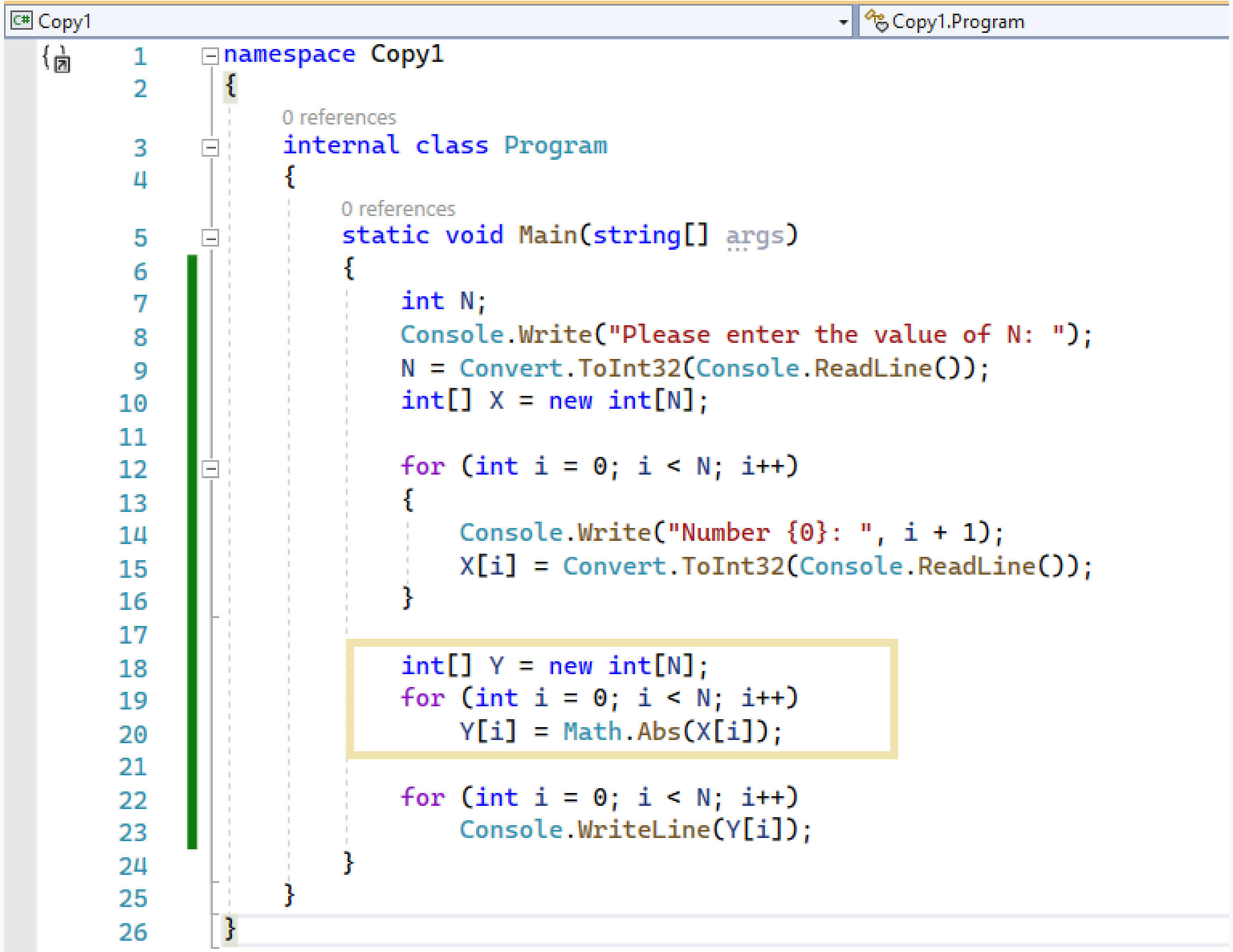
Length(X)	→	N
X[]	→	X[]
Y[]	→	Y[]
F(i)	→	ABS(X[i]



Copy absolute values

```
i=1..N
    Y[i]:=abs(X[i])
```

Source code 1 – Copy – Absolute values



```
namespace Copy1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int N;
            Console.Write("Please enter the value of N: ");
            N = Convert.ToInt32(Console.ReadLine());
            int[] X = new int[N];

            for (int i = 0; i < N; i++)
            {
                Console.Write("Number {0}: ", i + 1);
                X[i] = Convert.ToInt32(Console.ReadLine());
            }

            int[] Y = new int[N];
            for (int i = 0; i < N; i++)
                Y[i] = Math.Abs(X[i]);

            for (int i = 0; i < N; i++)
                Console.WriteLine(Y[i]);
        }
    }
}
```

Example 2 – Copy

Copy with transformation by parity: Copy even values without any transformation and copy the odd values multiplied by -1.

Specification:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{Z}^N$

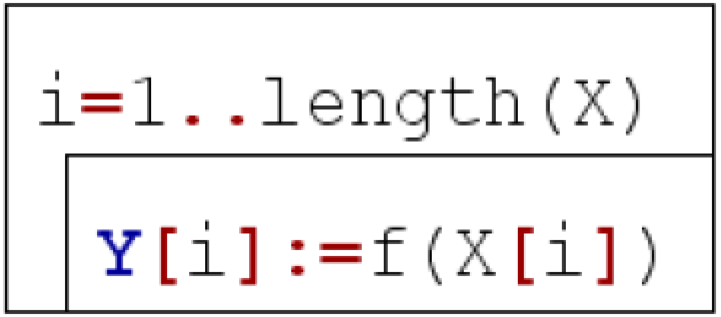
f: $\mathbb{Z} \rightarrow \mathbb{Z}$, Even: $\mathbb{Z} \rightarrow \mathbb{L}$, Even(x)=x|2

Output: $Y_{1..N} \in \mathbb{Z}^N$

Precondition: -

Postcondition: $\forall$ i $(1 \leq i \leq N)$: $Y_i = X_i$ and $X_i|2$ and $Y_i = -X_i$ and not $X_i \nmid 2$

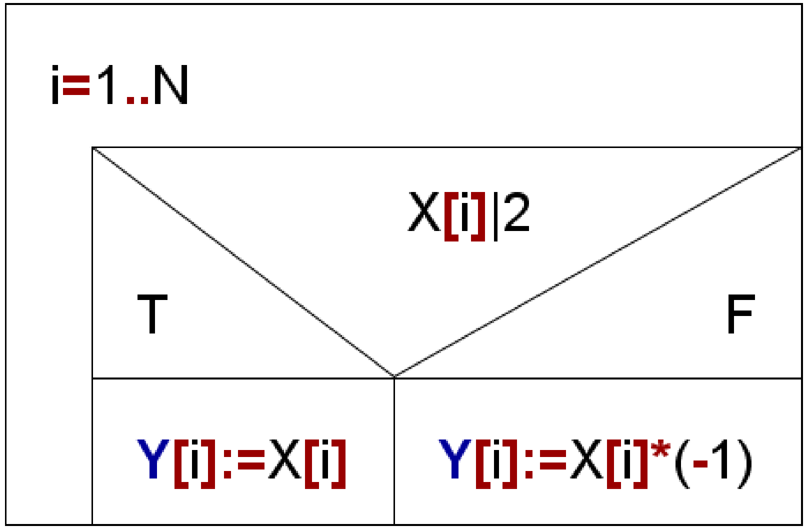
Algorithm



```
i=1..length(X)
    Y[i]:=f(X[i])
```

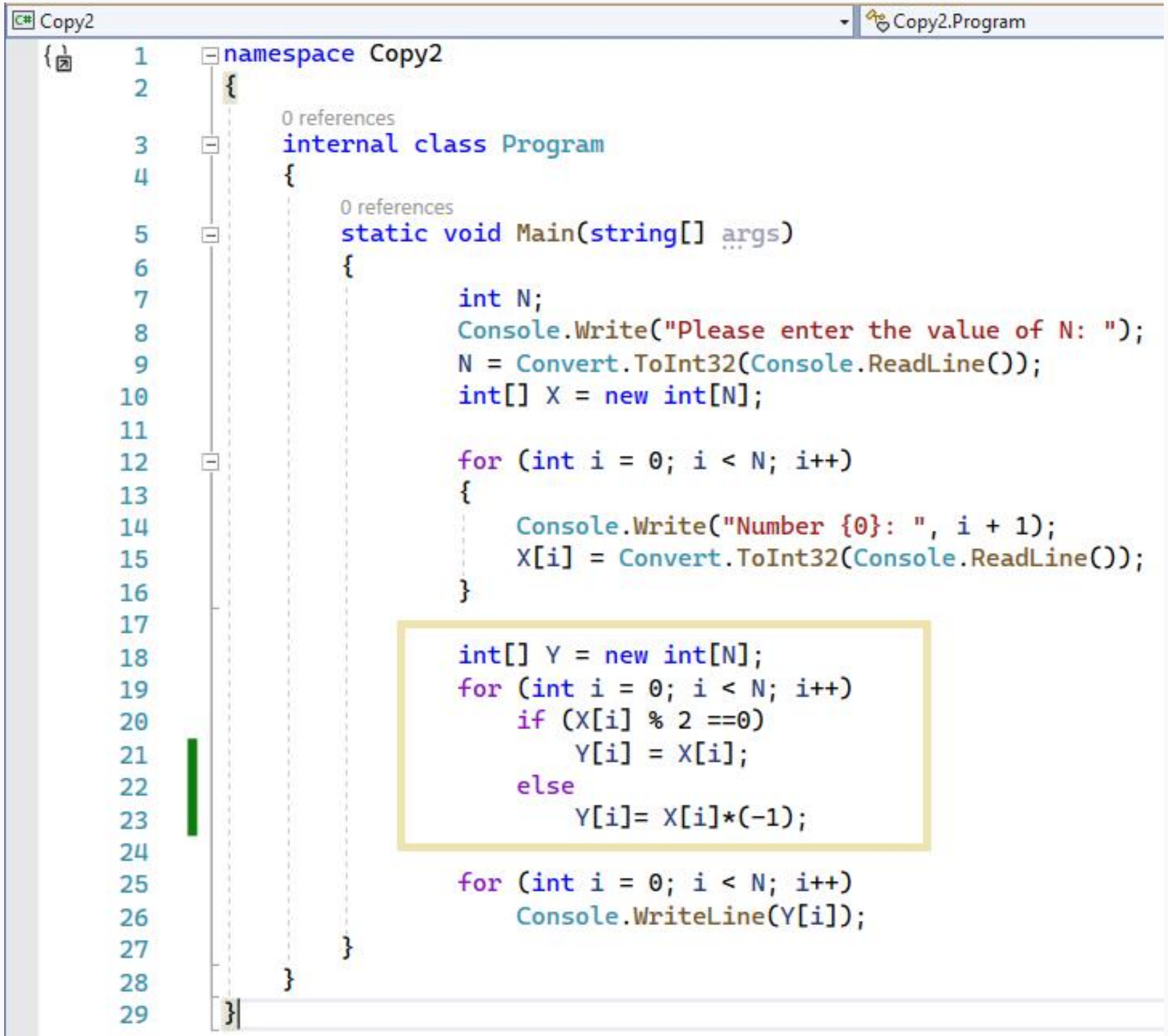
Length(X)	→	N
X[]	→	X[]
Y[]	→	Y[]
F(i)	→	$X_i|2$: X[i] $X_i \nmid 2$: - X[i]

Copy by parity



```
i=1..N
    X[i]|2
    T: Y[i]:=X[i]
    F: Y[i]:=X[i]*(-1)
```

Example 2 – Copy – Copy with transformation by parity



```
namespace Copy2
{
    internal class Program
    {
        static void Main(string[] args)
        {
                int N;
                Console.Write("Please enter the value of N: ");
                N = Convert.ToInt32(Console.ReadLine());
                int[] X = new int[N];

                for (int i = 0; i < N; i++)
                {
                    Console.Write("Number {0}: ", i + 1);
                    X[i] = Convert.ToInt32(Console.ReadLine());
                }

                int[] Y = new int[N];
                for (int i = 0; i < N; i++)
                    if (X[i] % 2 ==0)
                        Y[i] = X[i];
                    else
                        Y[i]= X[i]*(-1);

                for (int i = 0; i < N; i++)
                    Console.WriteLine(Y[i]);
        }
    }
}
```

Example 3 – Multiple item selection

Numbers that are divisible by 3: Select every numbers from the given array that are divisible by 3.

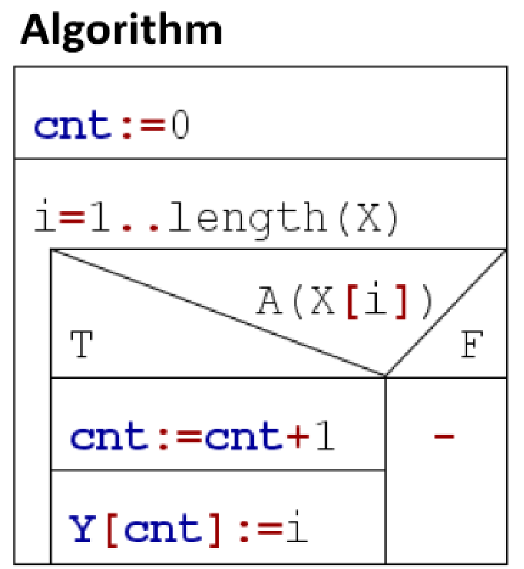
Specification:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{Z}^{N}$ $A: \mathbb{Z} \rightarrow \mathbb{L}$, $A(x)=x|3$

Output: $Cnt \in \mathbb{N}$, $Y_{1..Cnt} \in \mathbb{Z}^{Cnt}$

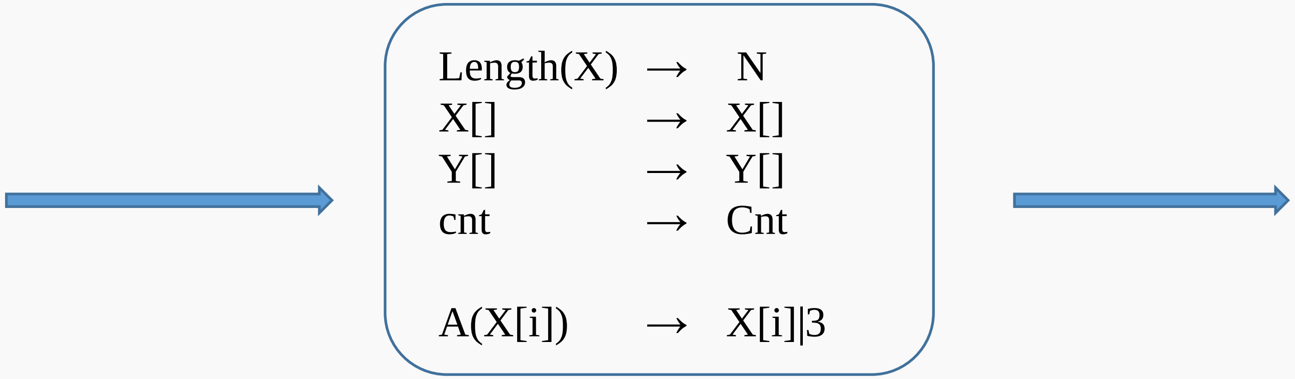
Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ X_i|3}}^{N} 1$ and $\forall i\ (1 \leq i \leq Cnt) : Y_i|3$ and $Y \subset X$

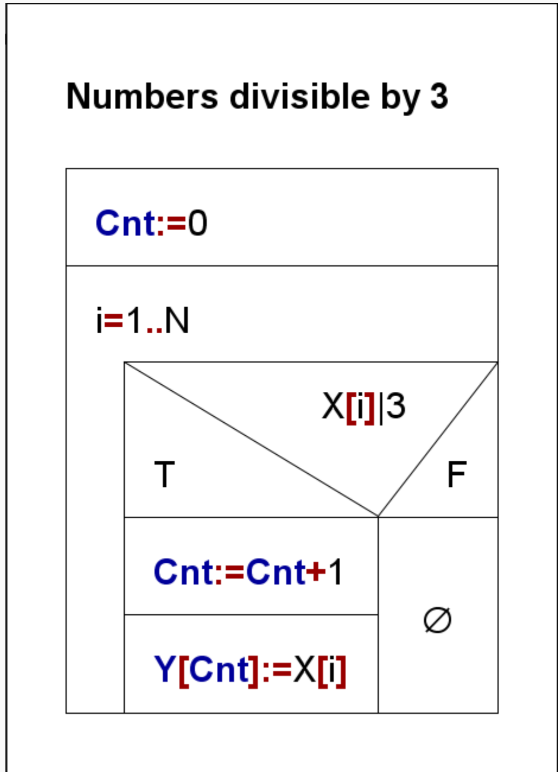


Algorithm

```
cnt:=0
i=1..length(X)
    A(X[i])
    T                 F
    cnt:=cnt+1        -
    Y[cnt]:=i
```



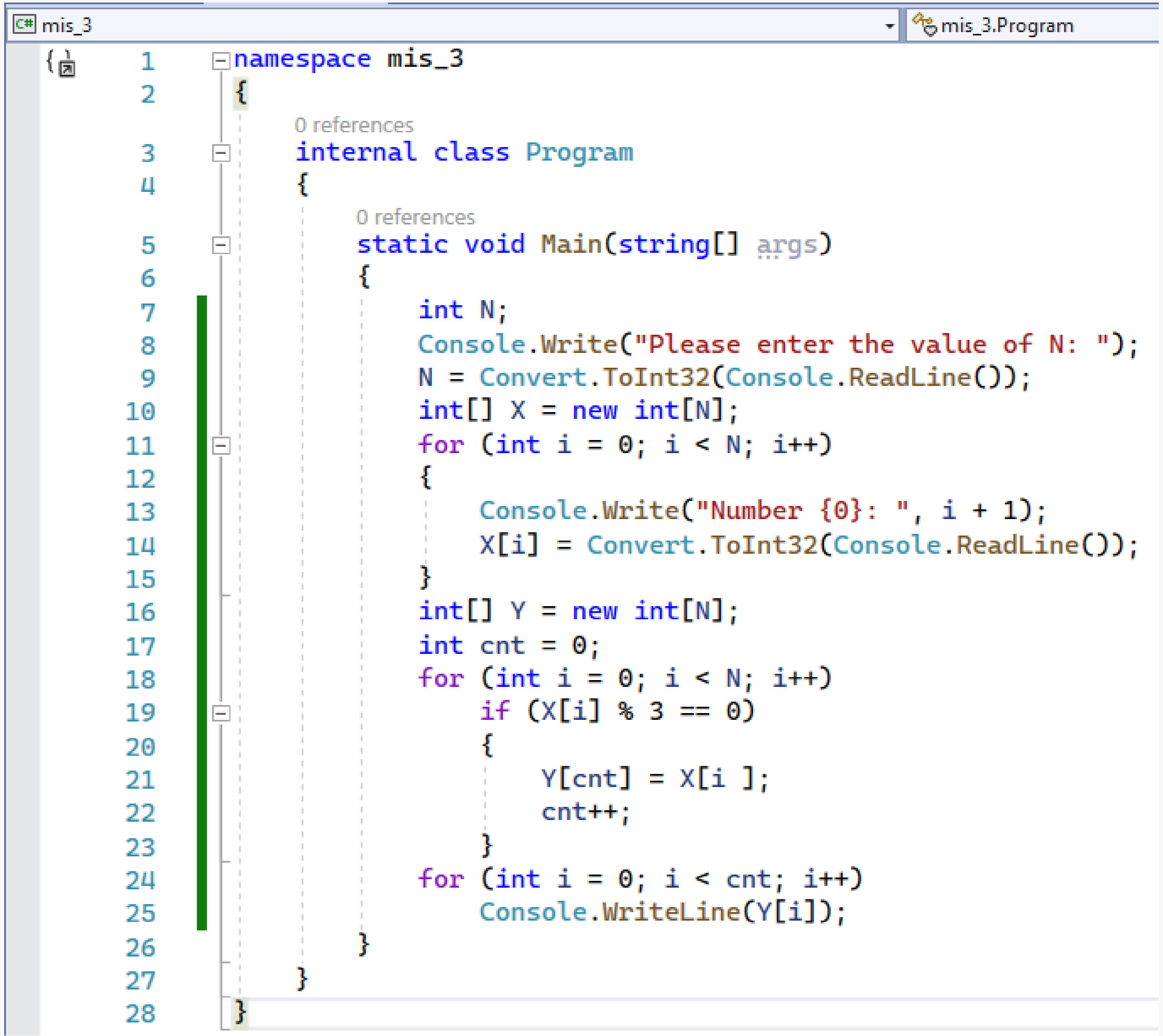
Length(X)	→	N
X[]	→	X[]
Y[]	→	Y[]
cnt	→	Cnt
A(X[i])	→	X[i]|3



Numbers divisible by 3

```
Cnt:=0
i=1..N
    X[i]|3
    T                 F
    Cnt:=Cnt+1        ∅
    Y[Cnt]:=X[i]
```

Example 3 – Multiple item selection – Numbers that are divisible by 3



```
namespace mis_3
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int N;
            Console.Write("Please enter the value of N: ");
            N = Convert.ToInt32(Console.ReadLine());
            int[] X = new int[N];
            for (int i = 0; i < N; i++)
            {
                Console.Write("Number {0}: ", i + 1);
                X[i] = Convert.ToInt32(Console.ReadLine());
            }
            int[] Y = new int[N];
            int cnt = 0;
            for (int i = 0; i < N; i++)
                if (X[i] % 3 == 0)
                {
                    Y[cnt] = X[i ];
                    cnt++;
                }
            for (int i = 0; i < cnt; i++)
                Console.WriteLine(Y[i]);
        }
    }
}
```

Example 4 – Multiple item selection

Negative numbers: Select every numbers from the given array that are less than 0.

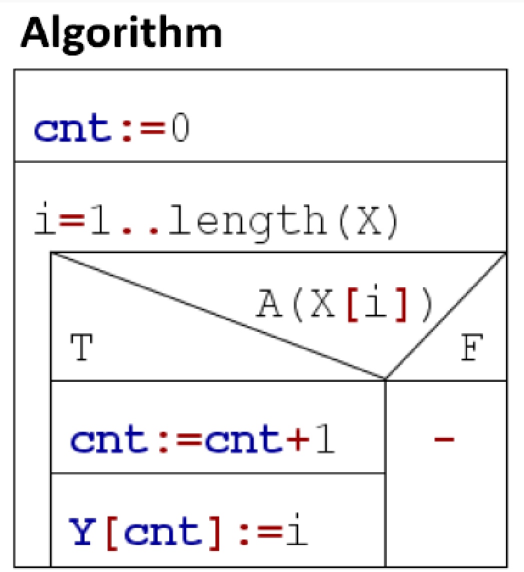
Specification:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{Z}^{N}$ $A: \mathbb{Z} \rightarrow \mathbb{L}$, $A(x)=x<0$

Output: $Cnt \in \mathbb{N}$, $Y_{1..Cnt} \in \mathbb{Z}^{Cnt}$

Precondition: –

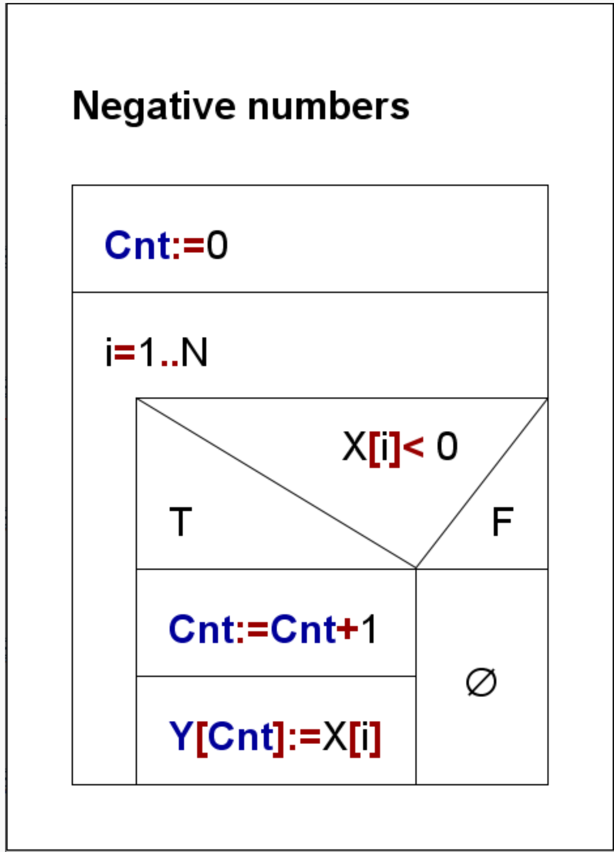
Postcondition: $Cnt = \sum_{\substack{i=1 \\ X_i<0}}^{N} 1$ and $\forall\, i\ (1 \leq i \leq Cnt) : Y_i < 0$ and $Y \subseteq X$



Algorithm

```
cnt:=0
i=1..length(X)
    A(X[i])
    T                     F
    cnt:=cnt+1            -
    Y[cnt]:=i
```

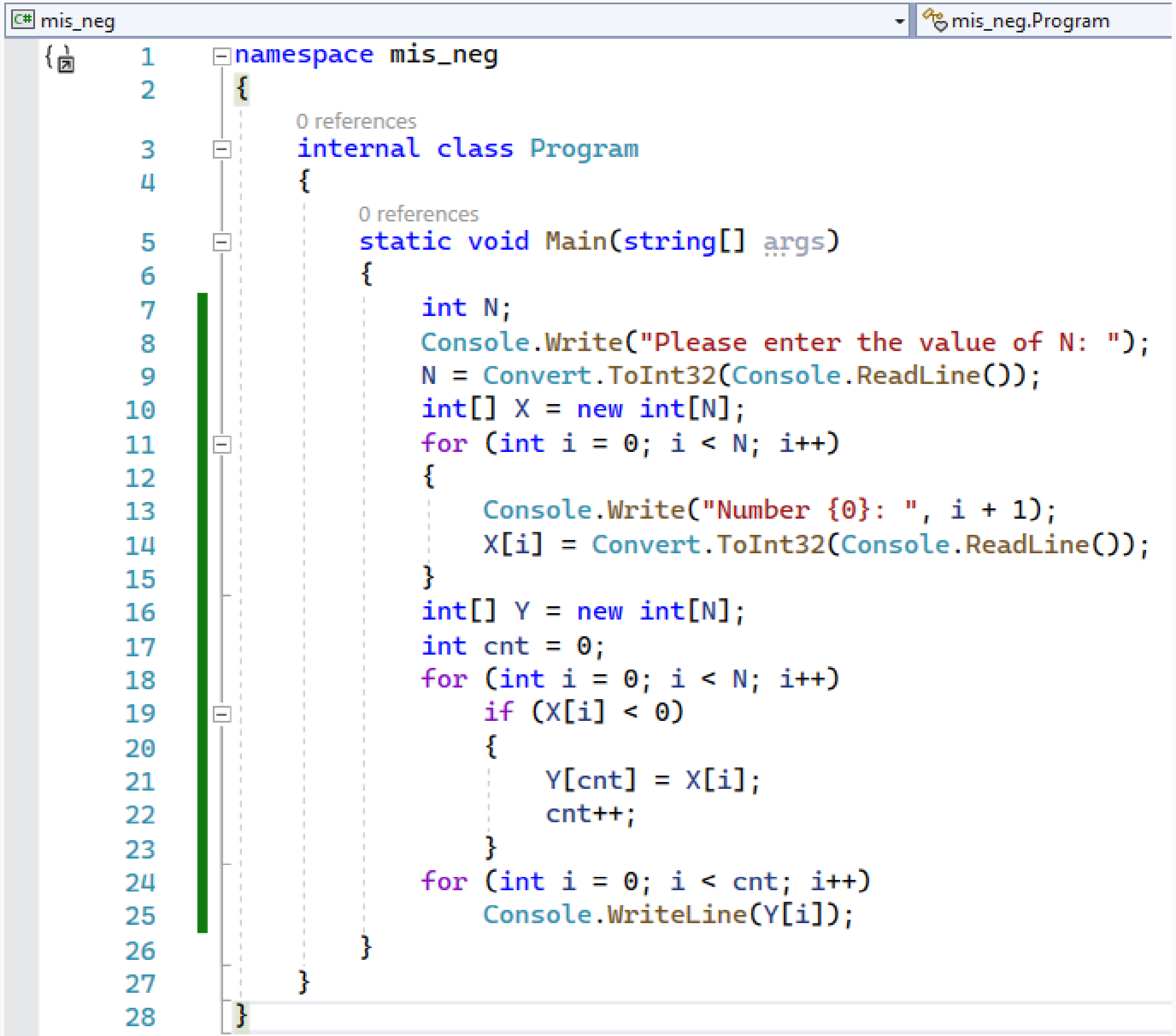
Length(X)	→	N
X[]	→	X[]
Y[]	→	Y[]
cnt	→	Cnt
A(X[i])	→	X[i]<0



Negative numbers

```
Cnt:=0
i=1..N
    X[i]< 0
    T                     F
    Cnt:=Cnt+1            ∅
    Y[Cnt]:=X[i]
```

Example 4 – Multiple item selection – Negative numbers



```
namespace mis_neg
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int N;
            Console.Write("Please enter the value of N: ");
            N = Convert.ToInt32(Console.ReadLine());
            int[] X = new int[N];
            for (int i = 0; i < N; i++)
            {
                Console.Write("Number {0}: ", i + 1);
                X[i] = Convert.ToInt32(Console.ReadLine());
            }
            int[] Y = new int[N];
            int cnt = 0;
            for (int i = 0; i < N; i++)
                if (X[i] < 0)
                {
                    Y[cnt] = X[i];
                    cnt++;
                }
            for (int i = 0; i < cnt; i++)
                Console.WriteLine(Y[i]);
        }
    }
}
```

Example 5 – Partitioning

Russian names: Please partition the element of X array - that contains Russian names - by gender

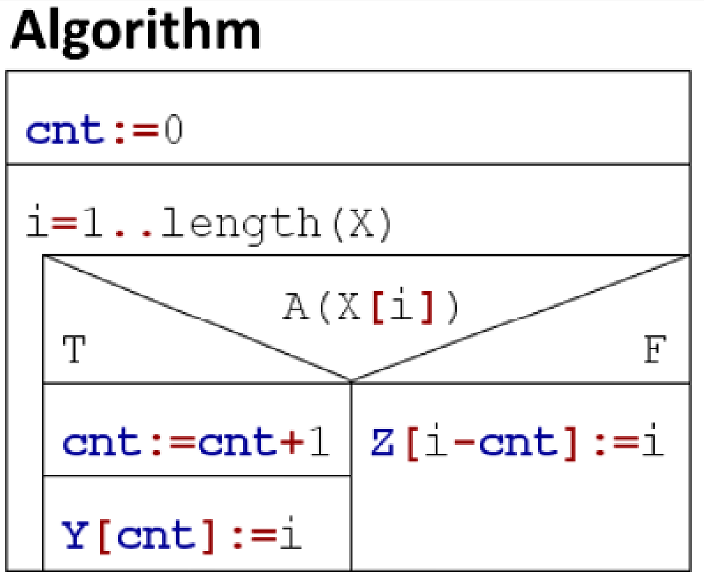
Specification:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{T}^{N}$ $A: \mathbb{T} \to \mathbb{L}$

Output: $Cnt \in \mathbb{N}$, $Y_{1..Cnt} \in \mathbb{T}^{Cnt}$, $Z_{1..N\text{-}Cnt} \in \mathbb{T}^{Cnt}$

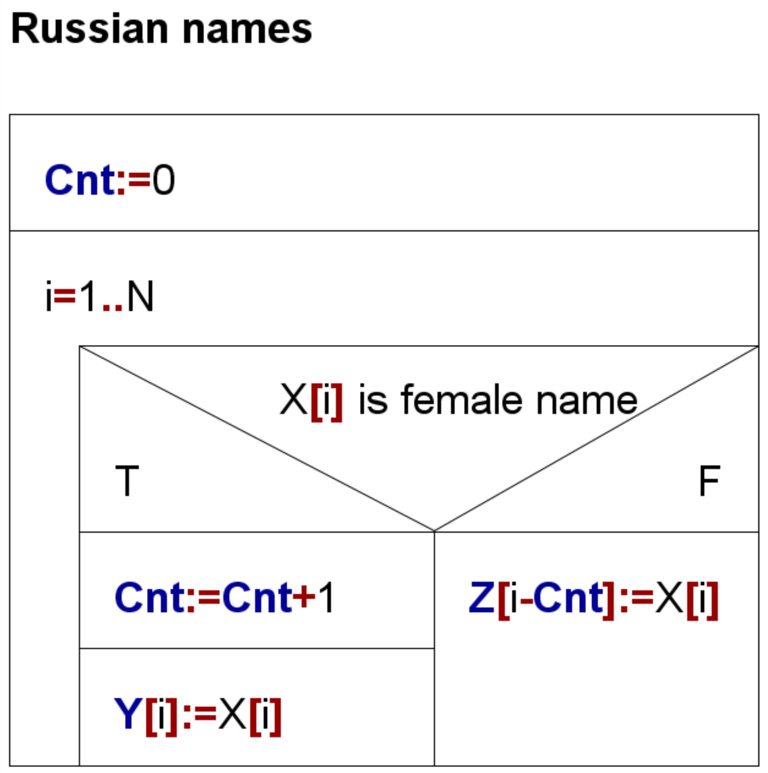
Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1\\ A(Xi)}}^{N} 1$ and $\forall$ i $(1 \leq i \leq Cnt)$: Yi is female name and $\forall$ i $(1 \leq i \leq N\text{-}Cnt)$: Zi is not female name and $Y \subseteq X$ and $Z \subseteq X$



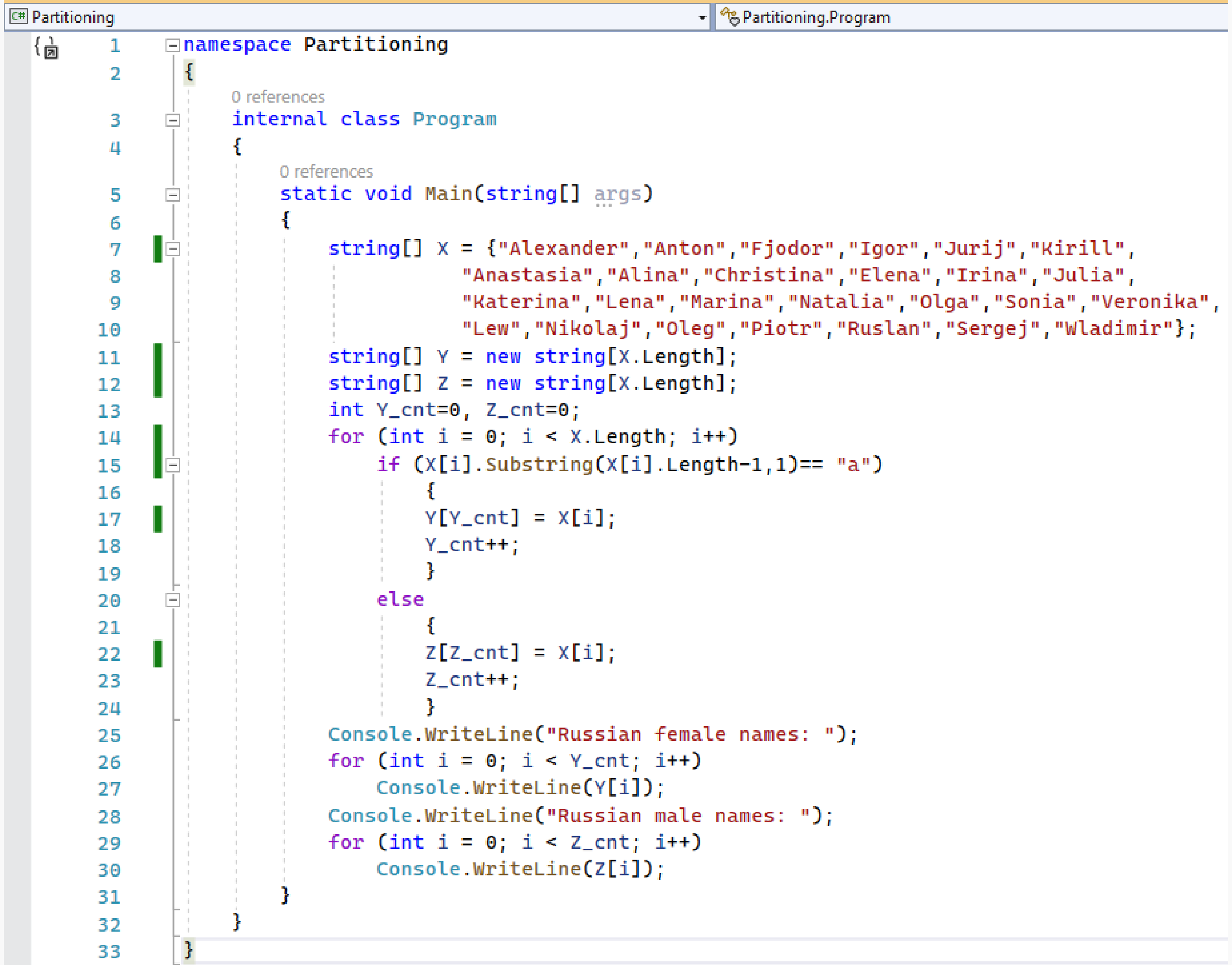
```
Algorithm
cnt:=0
i=1..length(X)
    A(X[i])
    T                     F
    cnt:=cnt+1            Z[i-cnt]:=i
    Y[cnt]:=i
```

Length(X)	→	N
X[]	→	X[]
Y[]	→	Y[]
Z[]	→	Z[]
A(X[i])	→	LastChar(X[i])='a'



```
Russian names
Cnt:=0
i=1..N
    X[i] is female name
    T                     F
    Cnt:=Cnt+1            Z[i-Cnt]:=X[i]
    Y[i]:=X[i]
```

Example 5 – Partitioning – Russian names



```
namespace Partitioning
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string[] X = {"Alexander","Anton","Fjodor","Igor","Jurij","Kirill",
                       "Anastasia","Alina","Christina","Elena","Irina","Julia",
                       "Katerina","Lena","Marina","Natalia","Olga","Sonia","Veronika",
                       "Lew","Nikolaj","Oleg","Piotr","Ruslan","Sergej","Wladimir"};
            string[] Y = new string[X.Length];
            string[] Z = new string[X.Length];
            int Y_cnt=0, Z_cnt=0;
            for (int i = 0; i < X.Length; i++)
                if (X[i].Substring(X[i].Length-1,1)== "a")
                    {
                    Y[Y_cnt] = X[i];
                    Y_cnt++;
                    }
                else
                    {
                    Z[Z_cnt] = X[i];
                    Z_cnt++;
                    }
            Console.WriteLine("Russian female names: ");
            for (int i = 0; i < Y_cnt; i++)
                Console.WriteLine(Y[i]);
            Console.WriteLine("Russian male names: ");
            for (int i = 0; i < Z_cnt; i++)
                Console.WriteLine(Z[i]);
        }
    }
}
```

Example 6 – Intersection

English names: We have 2 arrays. The first one contains common first names and the second one contains common surnames. Let’s list the names that can be used the first name as well surname.

Specification:

Input: $N, M \in \mathbb{N}$, $X_{1..N} \in \mathbb{T}^{N}$, $Y_{1..M} \in \mathbb{T}^{M}$

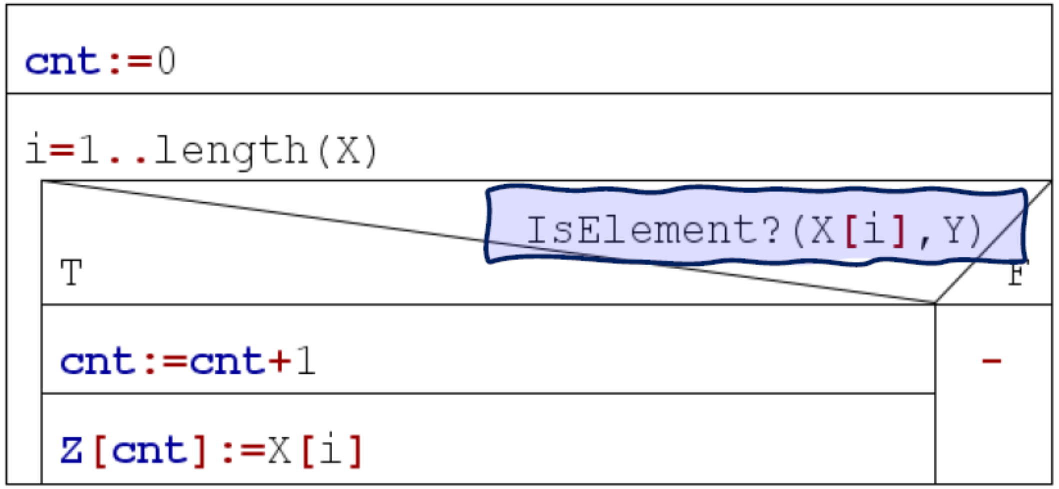
Output: $Cnt \in \mathbb{N}$, $Z_{1..MIN(N,M)} \in \mathbb{T}^{MIN(N,M)}$

Precondition: IsSet(X) and IsSet(Y)

Postcondition:

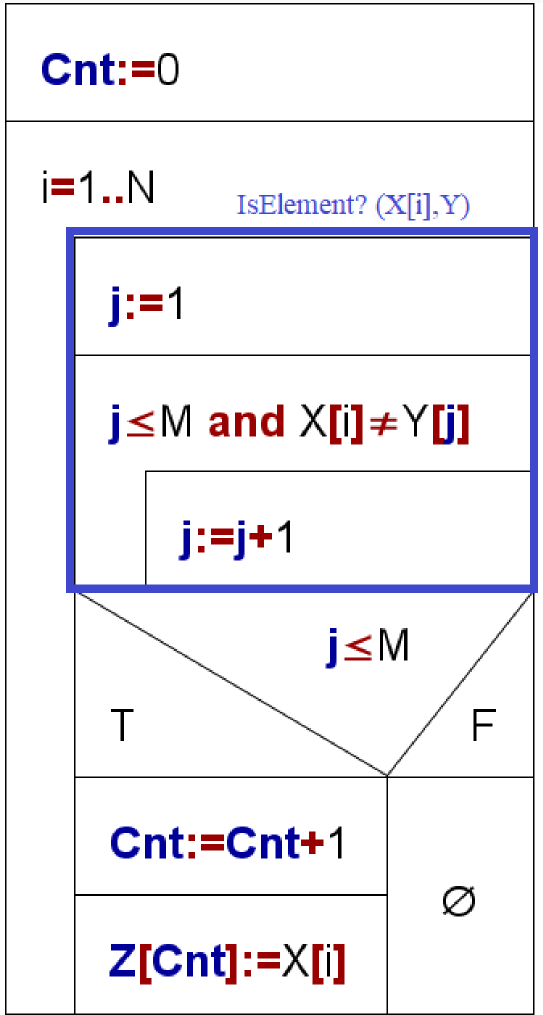
$$Cnt = \sum_{\substack{i=1\\ X_i \in Y}}^{N} 1 \text{ and } \forall\, i\ (1 \le i \le Cnt) : Z_i \in Y \text{ and IsSet}(Z)$$

Algorithm



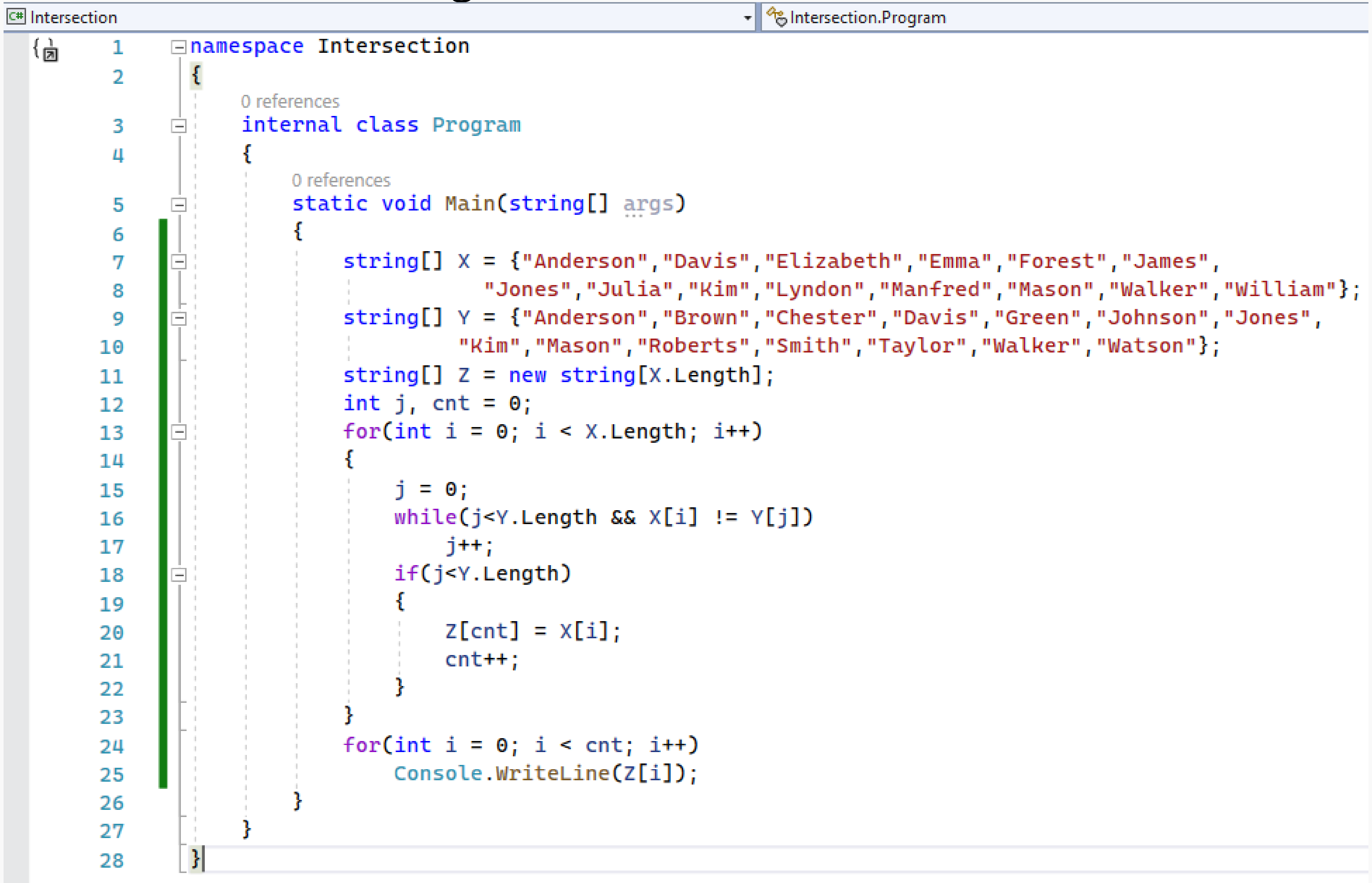
```
cnt:=0
i=1..length(X)
    IsElement?(X[i],Y)
    T:
        cnt:=cnt+1
        Z[cnt]:=X[i]
    F:
        -
```

length(X)	→	N
length(Y)	→	M
X[]	→	X[]
Y[]	→	Y[]
Z[]	→	Z[]
cnt	→	Cnt



```
Cnt:=0
i=1..N
    IsElement? (X[i],Y)
    j:=1
    j≤M and X[i]≠Y[j]
        j:=j+1
    j≤M
    T:
        Cnt:=Cnt+1
        Z[Cnt]:=X[i]
    F:
        ∅
```

Example 6 – Intersection – English names



```
namespace Intersection
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string[] X = {"Anderson","Davis","Elizabeth","Emma","Forest","James",
                       "Jones","Julia","Kim","Lyndon","Manfred","Mason","Walker","William"};
            string[] Y = {"Anderson","Brown","Chester","Davis","Green","Johnson","Jones",
                     "Kim","Mason","Roberts","Smith","Taylor","Walker","Watson"};
            string[] Z = new string[X.Length];
            int j, cnt = 0;
            for(int i = 0; i < X.Length; i++)
            {
                j = 0;
                while(j<Y.Length && X[i] != Y[j])
                    j++;
                if(j<Y.Length)
                {
                    Z[cnt] = X[i];
                    cnt++;
                }
            }
            for(int i = 0; i < cnt; i++)
                Console.WriteLine(Z[i]);
        }
    }
}
```

Example 7 – Union

Popular cars: We have 2 arrays. First one contains the most popular cars in Europe and the second one contains the most popular cars in Hungary. Let's list the popular cars in both area.

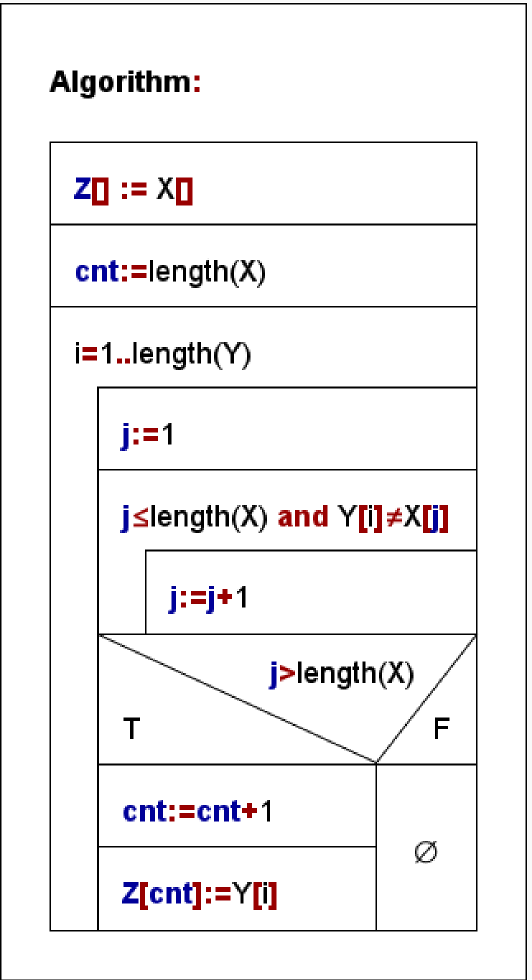
Specification:

Input: N, M $\in \mathbb{N}$, $X_{1..N} \in \mathbb{T}^{N}$, $Y_{1..M} \in \mathbb{T}^{M}$

Output: Cnt $\in \mathbb{N}$, $Z_{1..N+M} \in \mathbb{T}^{N+M}$

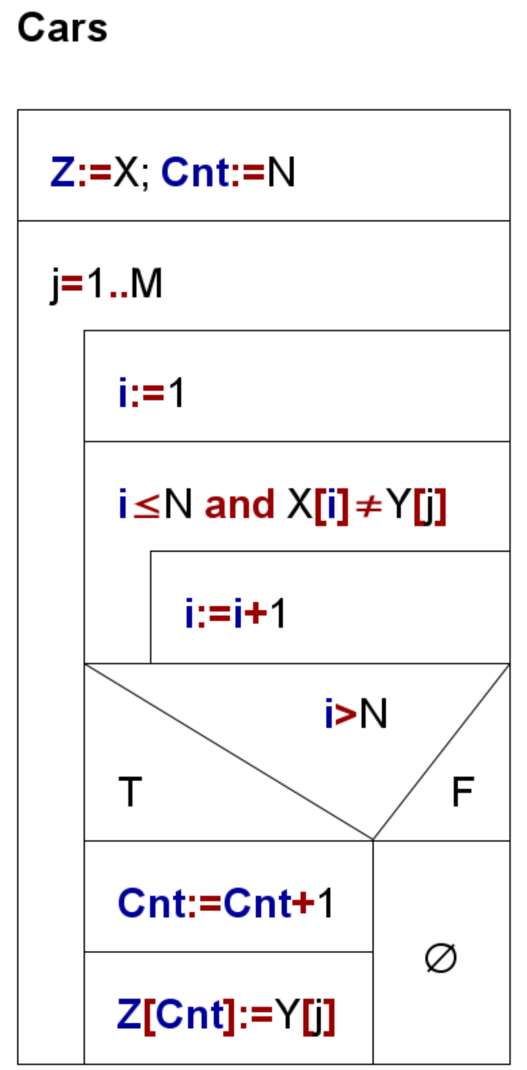
Precondition: IsSet(X) and IsSet(Y)

$$\text{Postcondition: } Cnt = N + \sum_{\substack{j=1 \\ Y_j \notin X}}^{N} 1 \ \text{ and } \forall\, i\ (1 \le i \le Cnt) : Z_i \in X \text{ or } Z_i \in Y \text{ and IsSet}(Z)$$



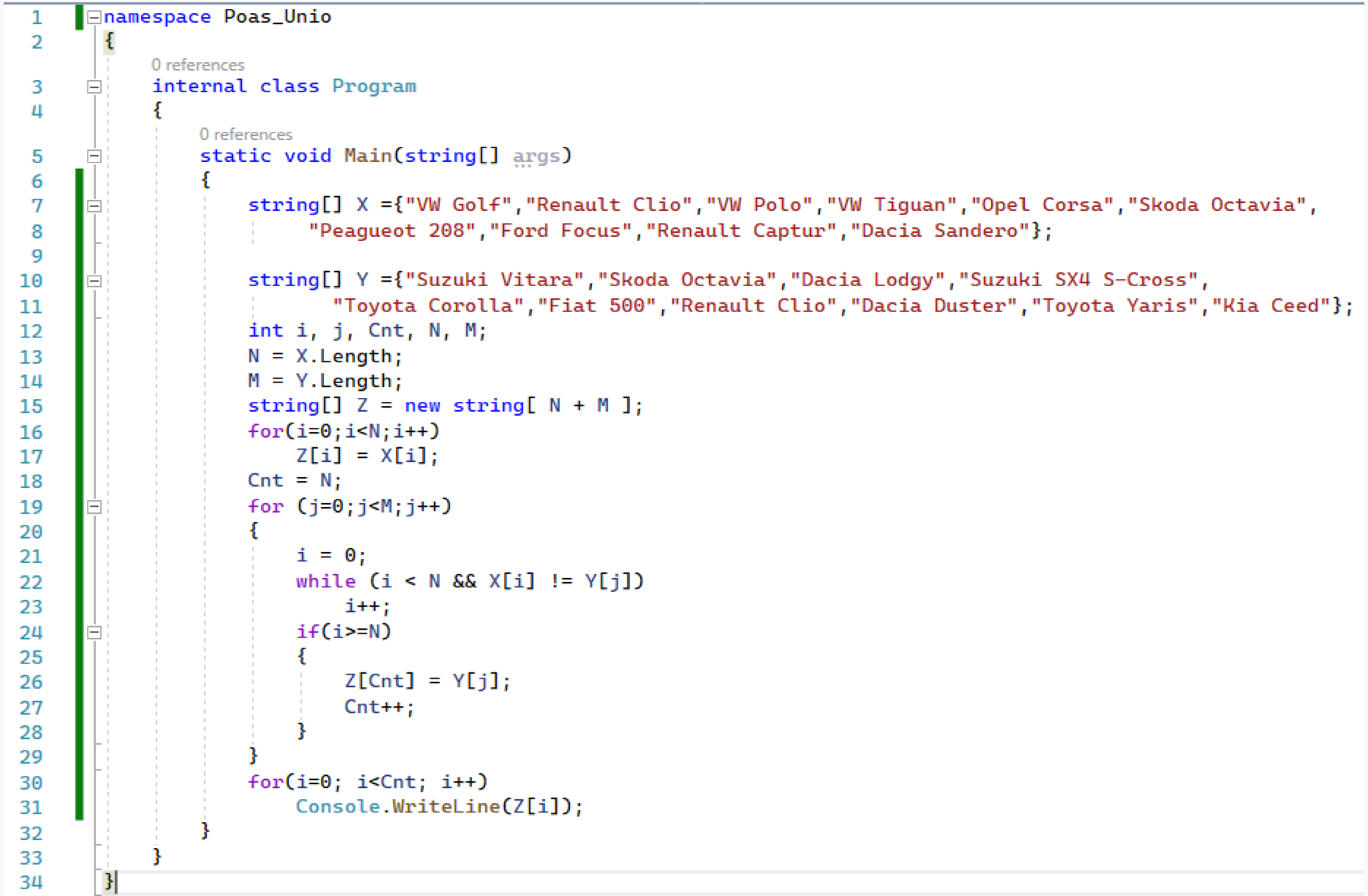
```
Algorithm:
Z[] := X[]
cnt:=length(X)
i=1..length(Y)
    j:=1
    j≤length(X) and Y[i]≠X[j]
        j:=j+1
    j>length(X)
    T                    F
    cnt:=cnt+1           ∅
    Z[cnt]:=Y[i]
```

length(X)	→	N
length(Y)	→	M
X[]	→	X[]
Y[]	→	Y[]
Z[]	→	Z[]
cnt	→	Cnt



```
Cars
Z:=X; Cnt:=N
j=1..M
    i:=1
    i≤N and X[i]≠Y[j]
        i:=i+1
    i>N
    T                    F
    Cnt:=Cnt+1           ∅
    Z[Cnt]:=Y[j]
```

Example 7 – Union – Popular cars



```
namespace Poas_Unio
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            string[] X ={"VW Golf","Renault Clio","VW Polo","VW Tiguan","Opel Corsa","Skoda Octavia",
                 "Peagueot 208","Ford Focus","Renault Captur","Dacia Sandero"};

            string[] Y ={"Suzuki Vitara","Skoda Octavia","Dacia Lodgy","Suzuki SX4 S-Cross",
                   "Toyota Corolla","Fiat 500","Renault Clio","Dacia Duster","Toyota Yaris","Kia Ceed"};
            int i, j, Cnt, N, M;
            N = X.Length;
            M = Y.Length;
            string[] Z = new string[ N + M ];
            for(i=0;i<N;i++)
                Z[i] = X[i];
            Cnt = N;
            for (j=0;j<M;j++)
            {
                i = 0;
                while (i < N && X[i] != Y[j])
                    i++;
                if(i>=N)
                {
                    Z[Cnt] = Y[j];
                    Cnt++;
                }
            }
            for(i=0; i<Cnt; i++)
                Console.WriteLine(Z[i]);
        }
    }
}
```