Patterns of Algorithms - Examples

Example 1 – Sequence calculation

Average of marks: We know the student's marks from a given subject. Let's calculate the average of marks.

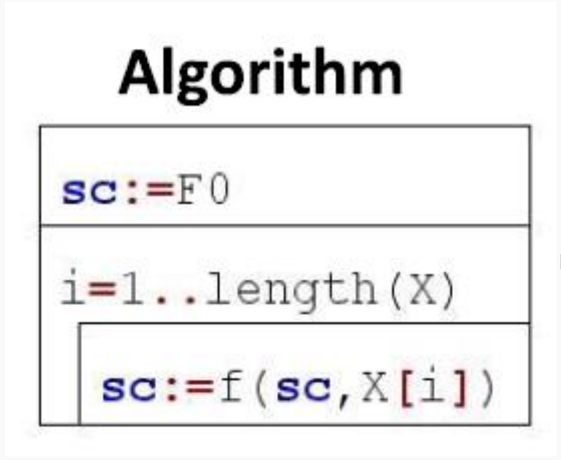
Specification:

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

Output: $AV \in \mathbb{R}$

Precondition: $\forall\, i\ (1 \le i \le N) : X_i \in [1..5]$

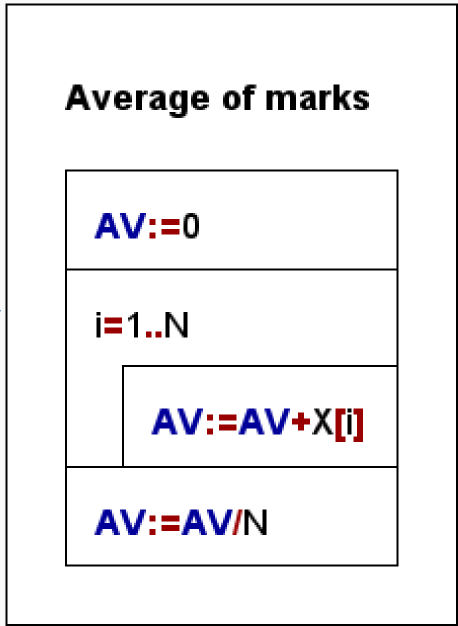
Postcondition: $AV = \dfrac{\sum_{i=1}^{N} X_i}{N}$



Algorithm

```
sc:=F0
i=1..length(X)
    sc:=f(sc,X[i])
```

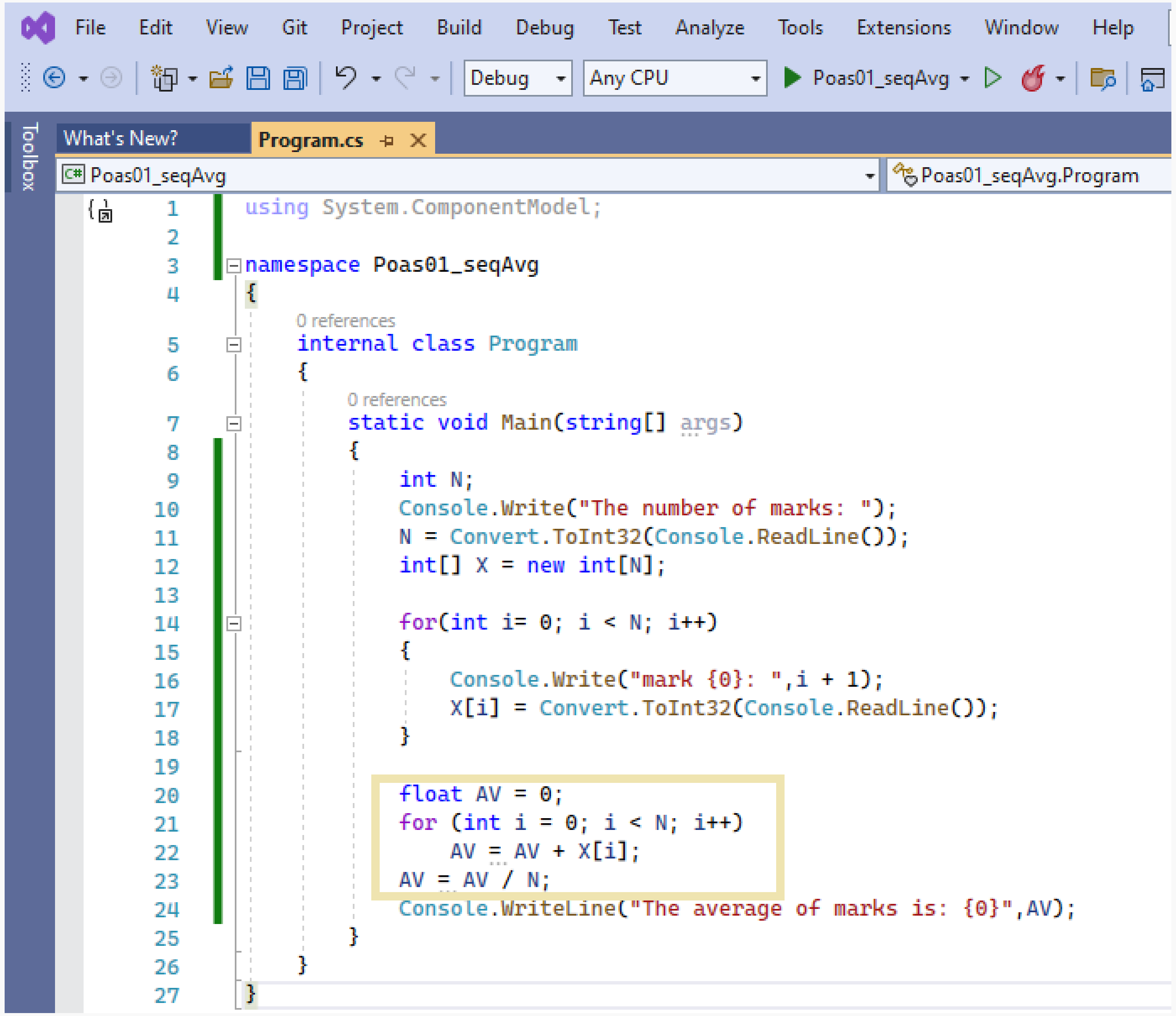
Length(X)	→	N
X[]	→	X[]
sc	→	AV
F0	→	0
F()	→	+



Average of marks

```
AV:=0
i=1..N
    AV:=AV+X[i]
AV:=AV/N
```

Source code 1 – Sequence calculation – Average of marks



```
using System.ComponentModel;

namespace Poas01_seqAvg
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int N;
            Console.Write("The number of marks: ");
            N = Convert.ToInt32(Console.ReadLine());
            int[] X = new int[N];

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

            float AV = 0;
            for (int i = 0; i < N; i++)
                AV = AV + X[i];
            AV = AV / N;
            Console.WriteLine("The average of marks is: {0}",AV);
        }
    }
}
```

Example 2 – Sequence calculation

Product of a and b by addition: Suppose our computer knows only one operation, that is addition. Let's calculate the product of **a** and **b** by addition.

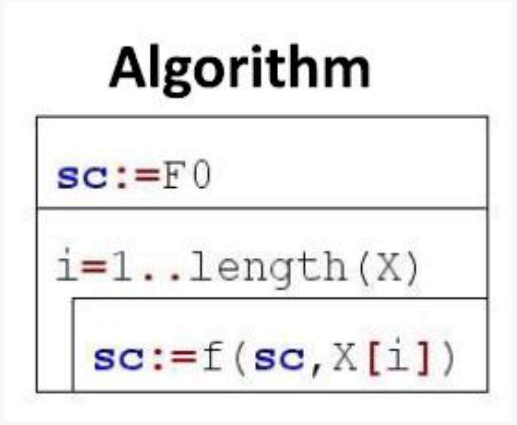
Specification:

Input: a, b ϵ $\mathbb{Z}$

Output: p ϵ $\mathbb{Z}$

Precondition: a, b $\neq$ 0

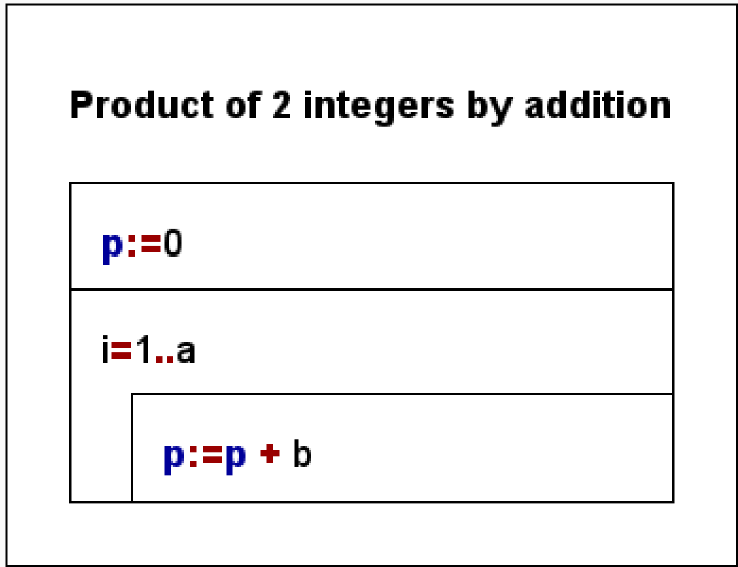
Postcondition: p = $\sum_{i=1}^{a} b$



Algorithm

```
sc:=F0
i=1..length(X)
    sc:=f(sc,X[i])
```

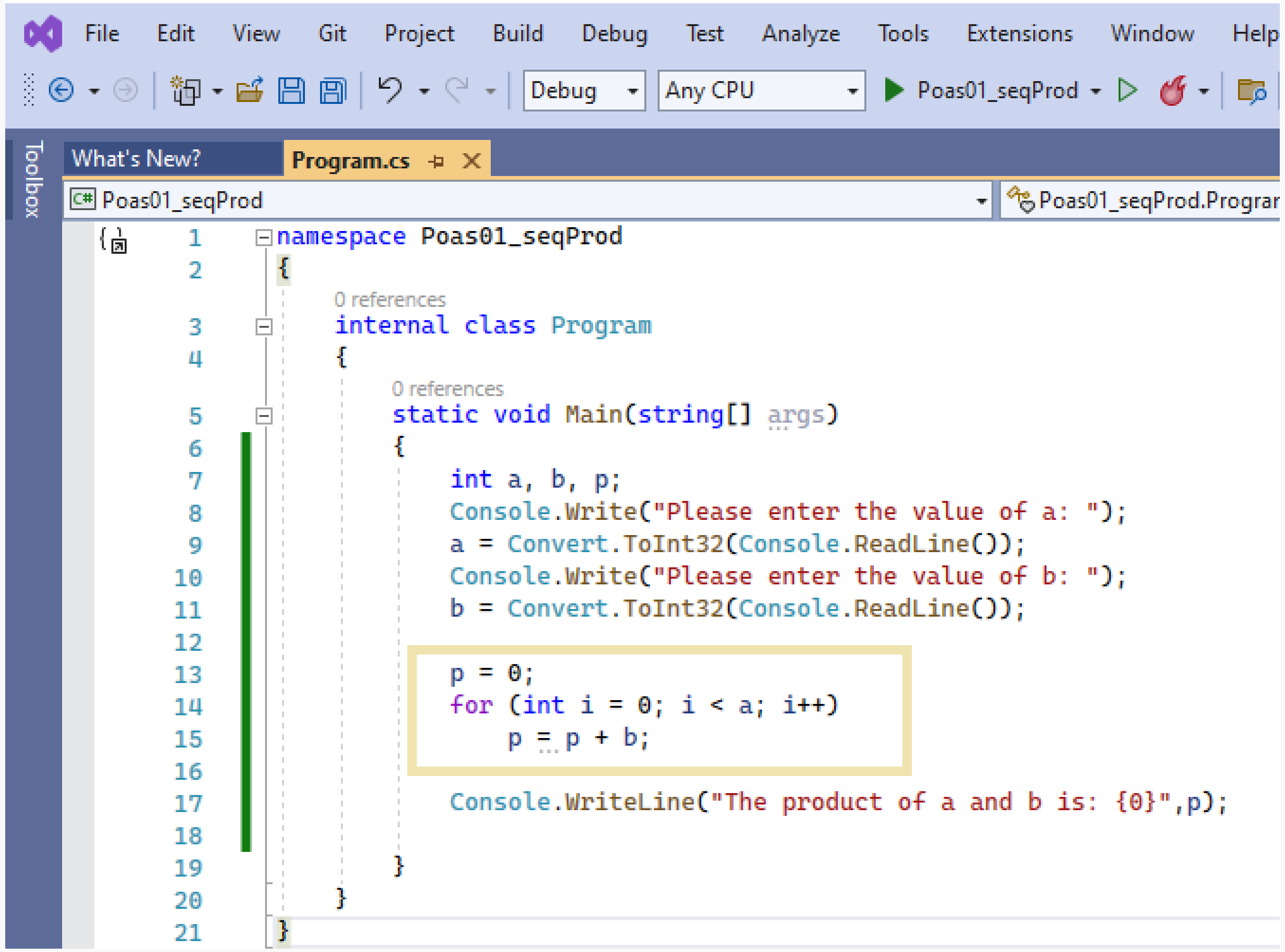
Length(X)	$\rightarrow$	a
X[]	$\rightarrow$	b
sc	$\rightarrow$	p
F0	$\rightarrow$	0
F()	$\rightarrow$	+



Product of 2 integers by addition

```
p:=0
i=1..a
    p:=p + b
```

Source code 2 – Sequence calculation – Product of a and b by addition



```
namespace Poas01_seqProd
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int a, b, p;
            Console.Write("Please enter the value of a: ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("Please enter the value of b: ");
            b = Convert.ToInt32(Console.ReadLine());

            p = 0;
            for (int i = 0; i < a; i++)
                p = p + b;

            Console.WriteLine("The product of a and b is: {0}",p);

        }
    }
}
```

Example 3 – Counting

The number of even numbers divisible by 7: Let’s count the number of even and divisible by 7 numbers from an array.

Specification:

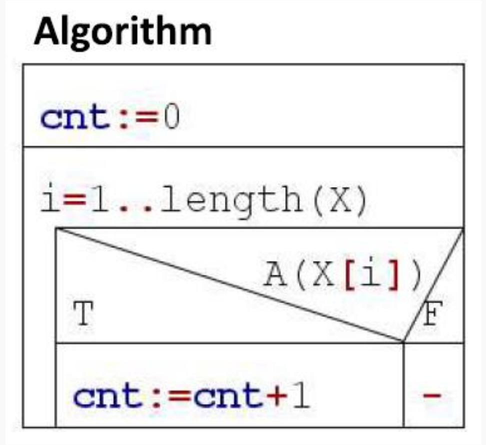
Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{N}^N$

$A: \mathbb{N} \rightarrow \mathbb{L}$ $A(x) := (x \mid 7 \text{ and } x \mid 2)$

Output: $cnt \in \mathbb{N}$

Precondition:

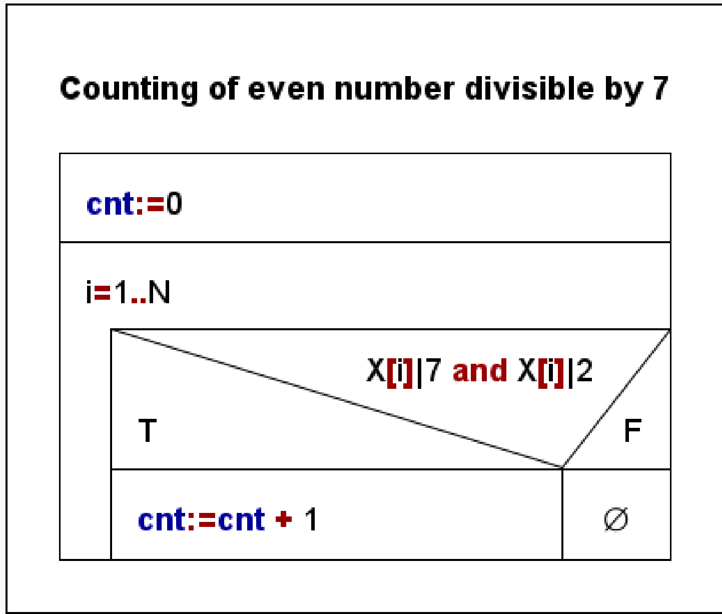
Postcondition: $cnt = \sum_{\substack{i=1 \\ A(X_i)}}^{N} 1$



Algorithm

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

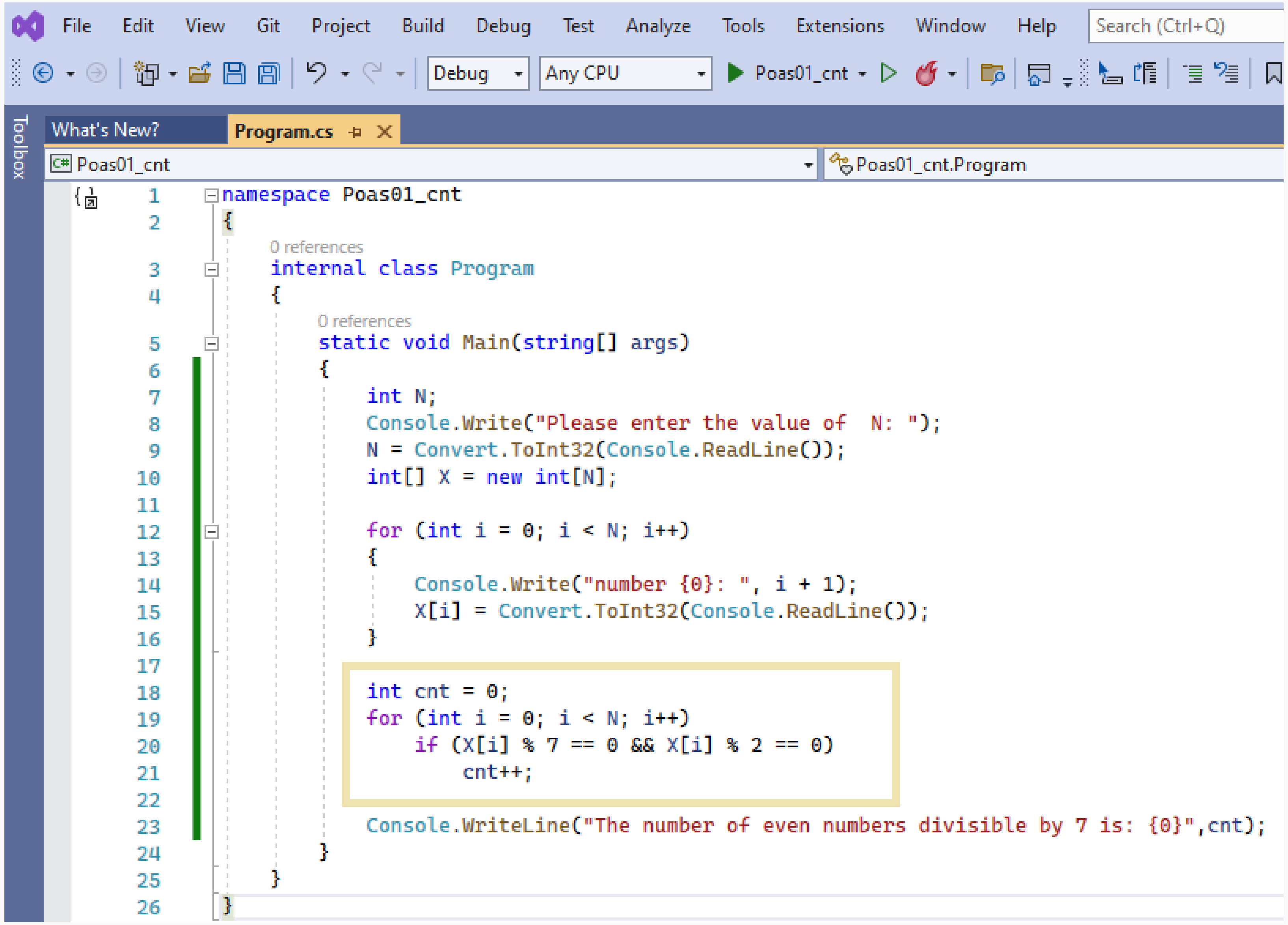
Length(X)	→	N
X[]	→	X[]
cnt	→	cnt
A(X[i])	→	x_i|7 and x_i|2



Counting of even number divisible by 7

```
cnt:=0
i=1..N
    X[i]|7 and X[i]|2
    T: cnt:=cnt + 1
    F: ∅
```

Source code 3 – Counting – The number of even numbers divisible by 7



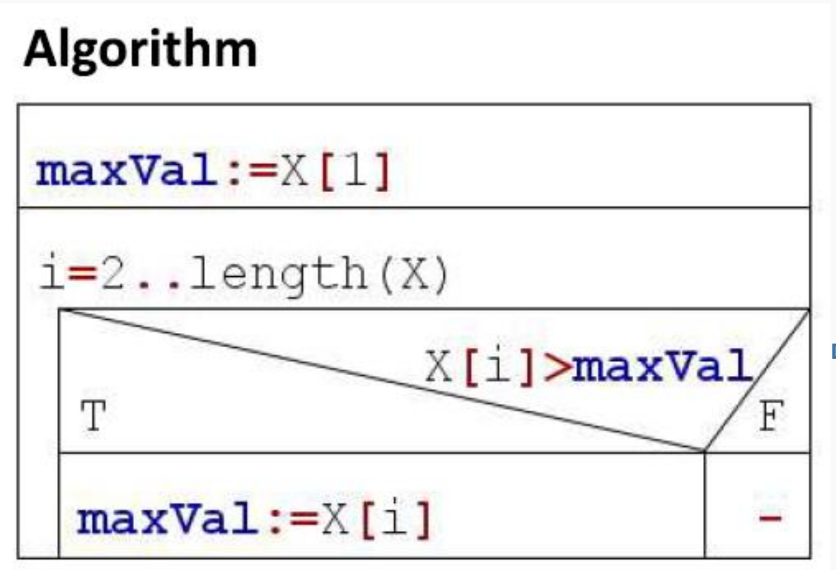
```csharp
namespace Poas01_cnt
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int N;
            Console.Write("Please enter the value of  N: ");
            N = Convert.ToInt32(Console.ReadLine());
            int[] X = new int[N];

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

            int cnt = 0;
            for (int i = 0; i < N; i++)
                if (X[i] % 7 == 0 && X[i] % 2 == 0)
                    cnt++;

            Console.WriteLine("The number of even numbers divisible by 7 is: {0}",cnt);
        }
    }
}
```

Example 4 – Maximum selection

The longest name: There is a list with the name of students. Let's select the longest name from this list.
Specification:

Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{S}^N$
Output: longest $\in \mathbb{S}$

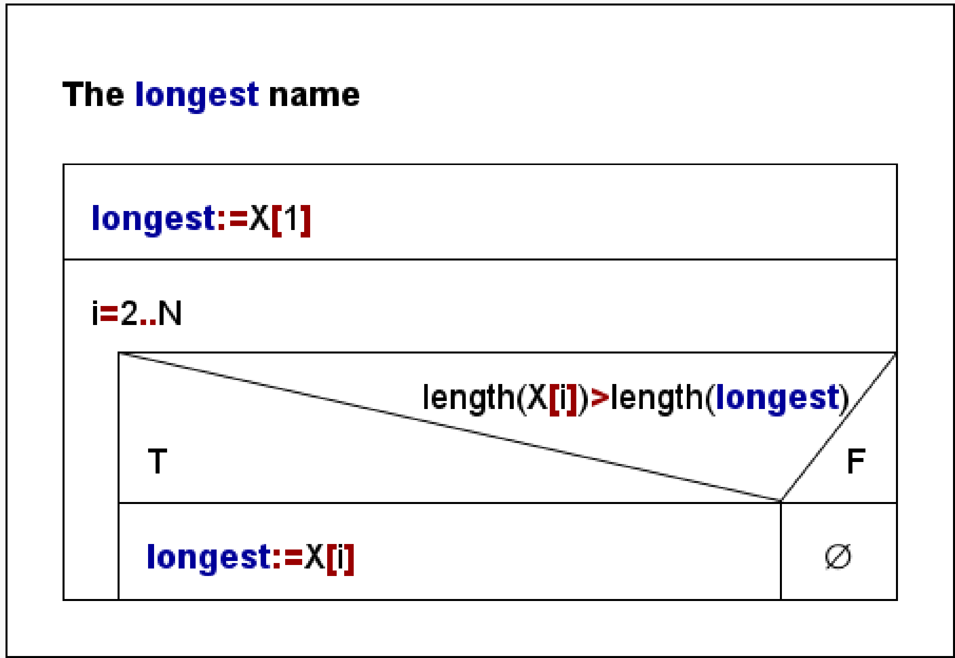
Precondition: $N > 0$ and $\exists\, i\ (1 \le i \le N) : \text{length}(X_i) > 0$
Postcondition: longest $\in X$ and $\exists\, i\ (1 \le i \le N)$: longest $= X_i$ and $\forall i\ (1 \le i \le N)$:longest $\ge \text{length}(X_i)$



Algorithm

```
maxVal:=X[1]
i=2..length(X)
    X[i]>maxVal
    T: maxVal:=X[i]
    F: -
```

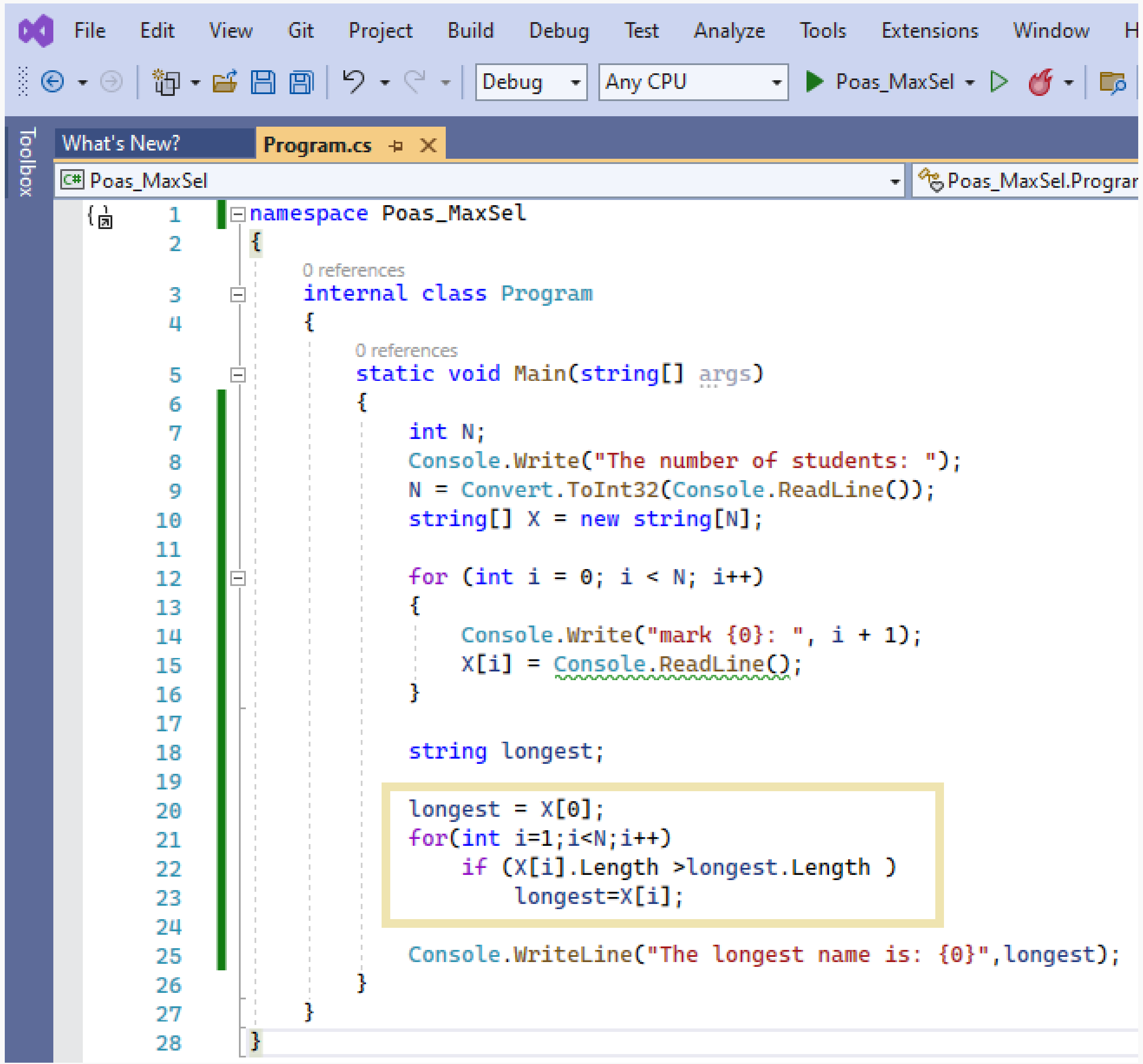
Length(X)	→	N
X[]	→	X[]
maxVal	→	longest
A(X[i])	→	$\text{length}(x_i) \le$ longest



The longest name

```
longest:=X[1]
i=2..N
    length(X[i])>length(longest)
    T: longest:=X[i]
    F: ∅
```

Source code 4 – Maximum selection – The longest name



```
namespace Poas_MaxSel
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int N;
            Console.Write("The number of students: ");
            N = Convert.ToInt32(Console.ReadLine());
            string[] X = new string[N];

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

            string longest;

            longest = X[0];
            for(int i=1;i<N;i++)
                if (X[i].Length >longest.Length )
                    longest=X[i];

            Console.WriteLine("The longest name is: {0}",longest);
        }
    }
}
```

Example 5 – Search

Square number: Let's looking for a square number within a sequence of numbers.

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

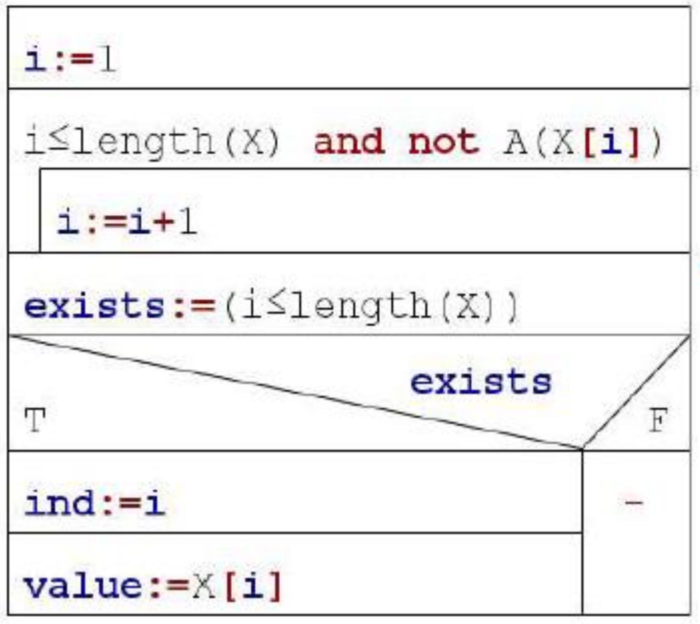
Output: $Exists \in \mathbb{L}$, $Ind \in \mathbb{N}$, $Val \in \mathbb{N}$

Precondition: $\forall i \; (1 \leq i \leq N) : X_i \geq 0$

Postcondition: $Exists = (\exists \; ind \; (1 \leq ind \leq N) : sqrt(X_{ind})$ is Integer

and $Exists \rightarrow 1 \leq ind \leq N$ and $sqrt(X_{ind})$ is Integer

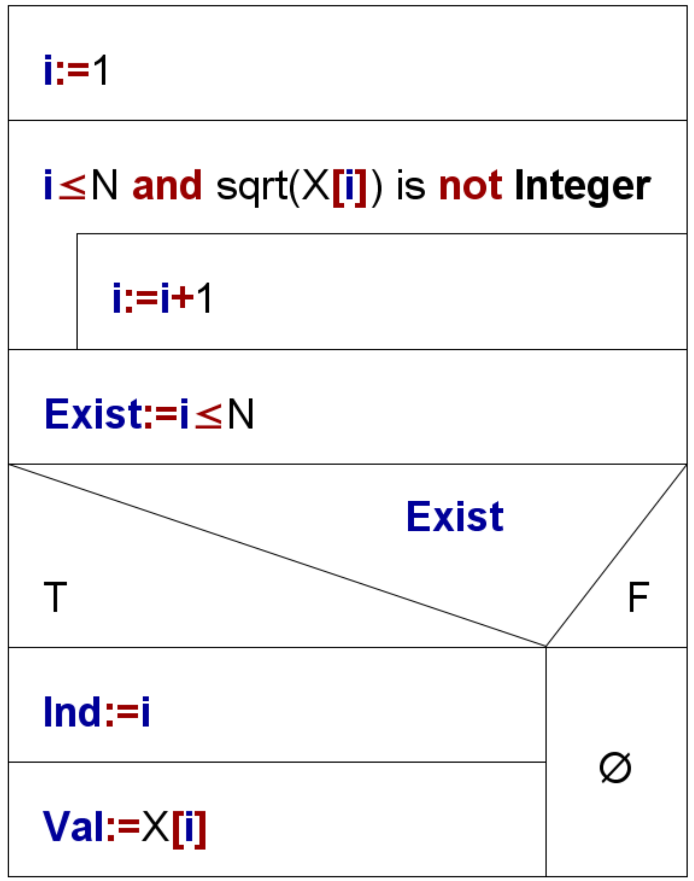
Algorithm



```
i:=1
i≤length(X) and not A(X[i])
    i:=i+1
exists:=(i≤length(X))
exists
  T:
    ind:=i
    value:=X[i]
  F:
    –
```

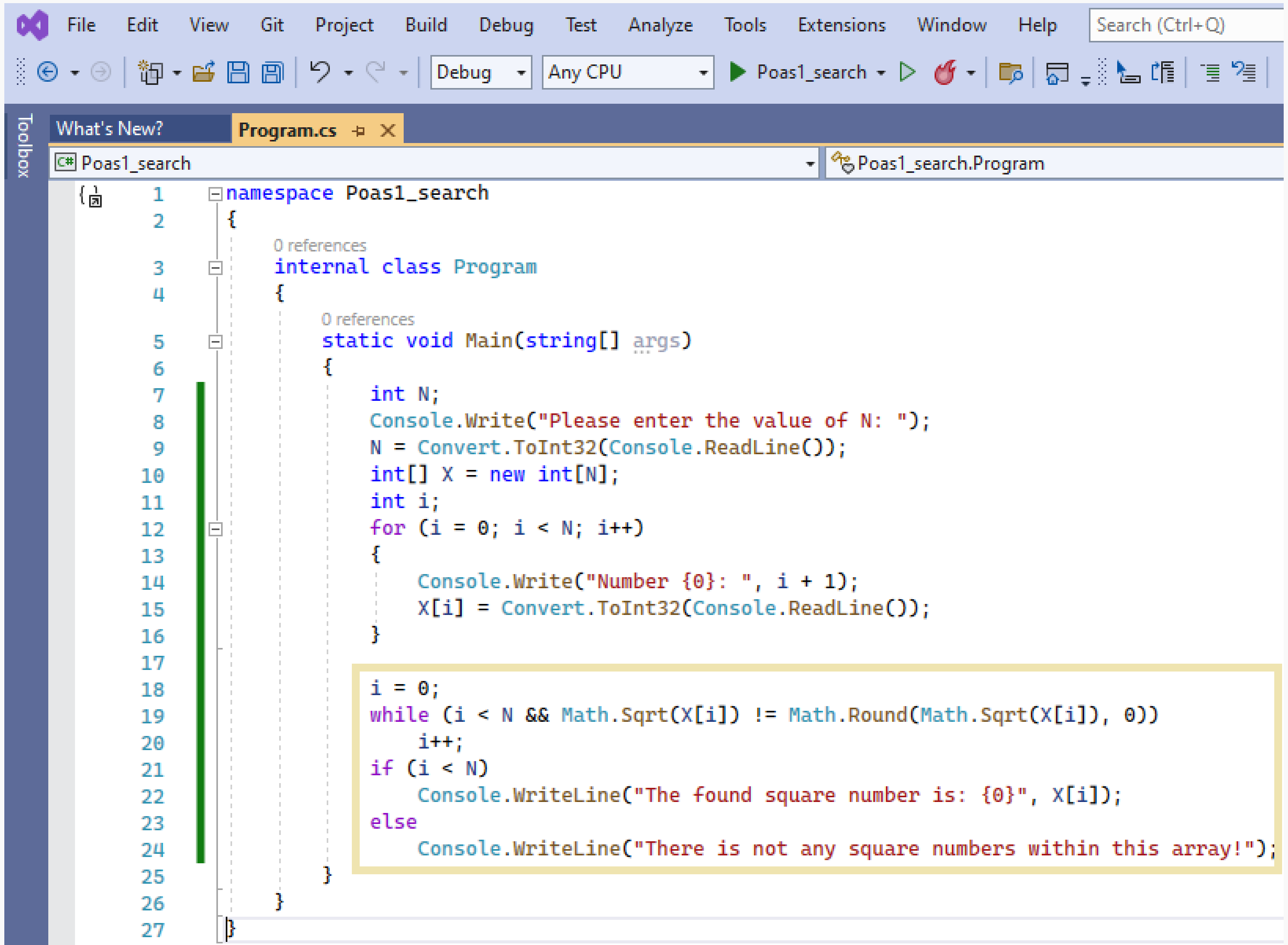
Length(X)	→	N
X[]	→	X[]
existx	→	Exists
A(X[i])	→	sqrt(X[i])=int(sqrt(X[i]))

Search Square Number



```
i:=1
i≤N and sqrt(X[i]) is not Integer
    i:=i+1
Exist:=i≤N
Exist
  T:
    Ind:=i
    Val:=X[i]
  F:
    ∅
```

Source code 5 – Search – Square number



```
namespace Poas1_search
{
    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];
            int i;
            for (i = 0; i < N; i++)
            {
                Console.Write("Number {0}: ", i + 1);
                X[i] = Convert.ToInt32(Console.ReadLine());
            }

            i = 0;
            while (i < N && Math.Sqrt(X[i]) != Math.Round(Math.Sqrt(X[i]), 0))
                i++;
            if (i < N)
                Console.WriteLine("The found square number is: {0}", X[i]);
            else
                Console.WriteLine("There is not any square numbers within this array!");
        }
    }
}
```

Example 6 – Decision

Divisible by 3: Let's make a decision is there any number that is divisible by 3 within a sequence of numbers.

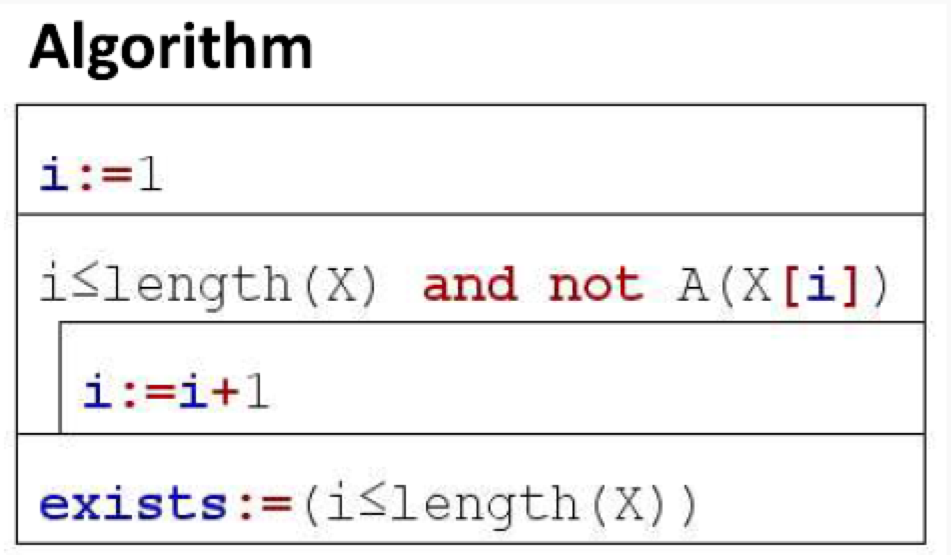
Specification:

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

Output: $Exists \in \mathbb{L}$

Precondition: –

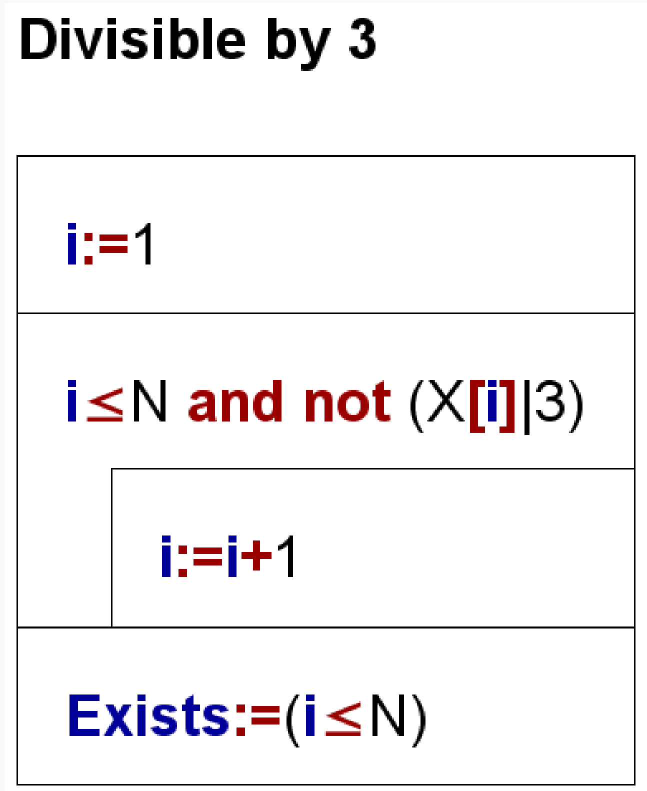
Postcondition: $Exists = \exists\, i\ (1 \leq i \leq N) : Xi|3$



Algorithm

```
i:=1
i≤length(X) and not A(X[i])
    i:=i+1
exists:=(i≤length(X))
```

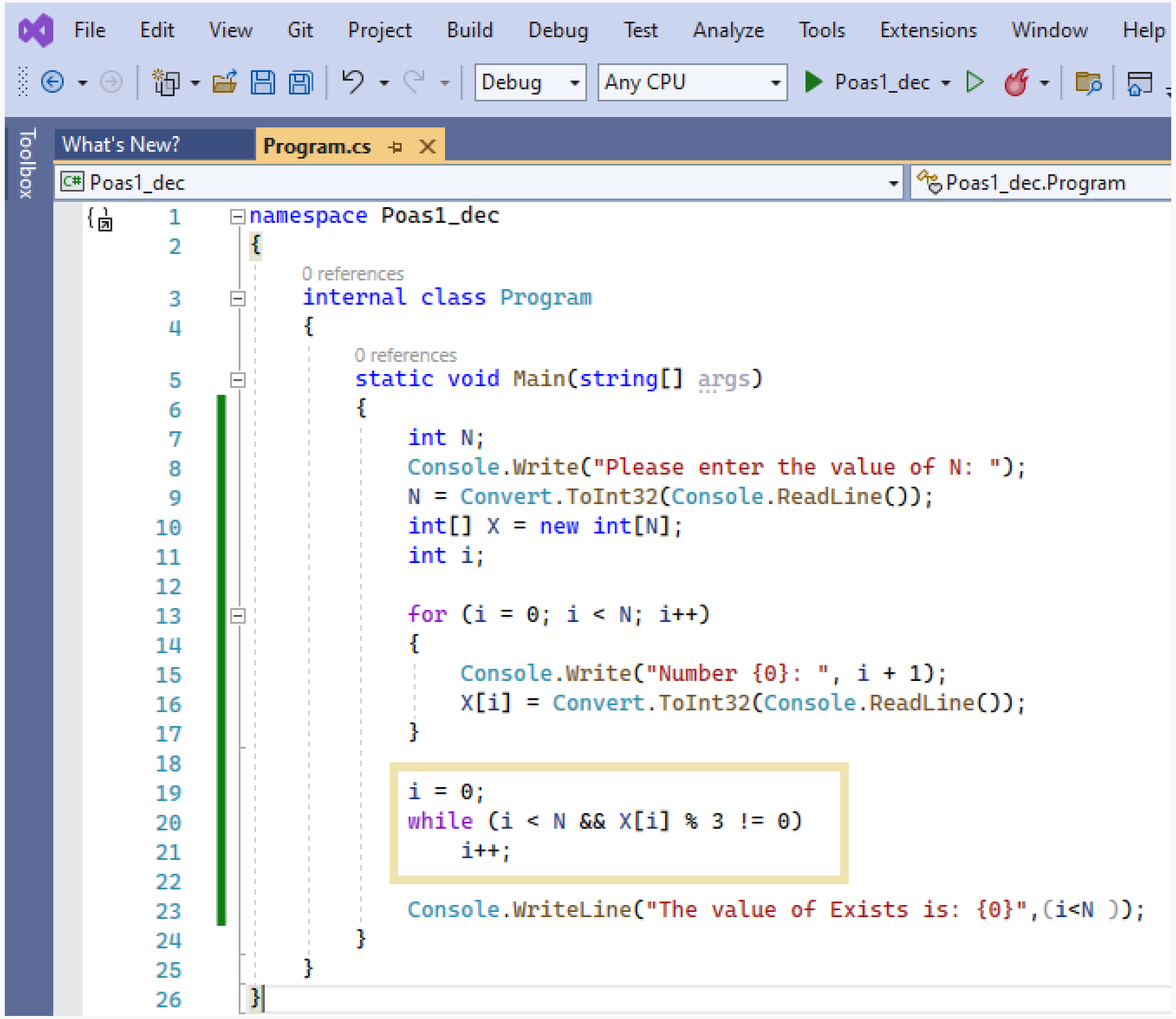
Length(X)	→	N
X[]	→	X[]
existx	→	Exists
A(X[i])	→	Xi|3



Divisible by 3

```
i:=1
i≤N and not (X[i]|3)
    i:=i+1
Exists:=(i≤N)
```

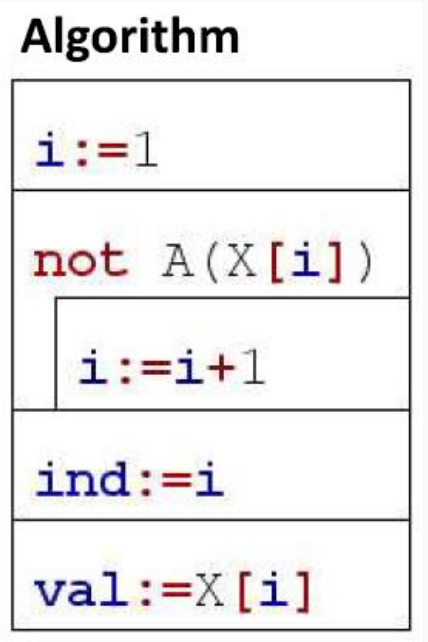
Source code 6 – Decision – Divisible by 3



```
namespace Poas1_dec
{
    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];
            int i;

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

            i = 0;
            while (i < N && X[i] % 3 != 0)
                i++;

            Console.WriteLine("The value of Exists is: {0}",(i<N ));
        }
    }
}
```

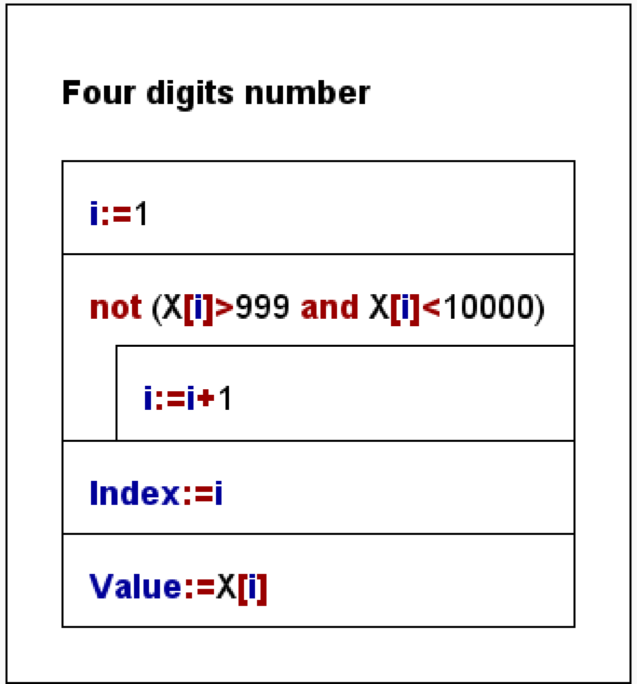
Example 7 – Selection

Four digits number: Let's select the first four digits number from a sequence of numbers.
Specification:
Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{N}^N$, $A: \mathbb{N} \to \mathbb{L}$
Output: $Index \in \mathbb{N}$, $Value \in \mathbb{N}$
Precondition: $\mathbb{N} > 0$ and $\exists\, i\ (1 \leq i \leq N): A(X_i)$
Postcondition: $1 \leq Index \leq N$ and $1000 \leq X_{Index} \leq 9999$



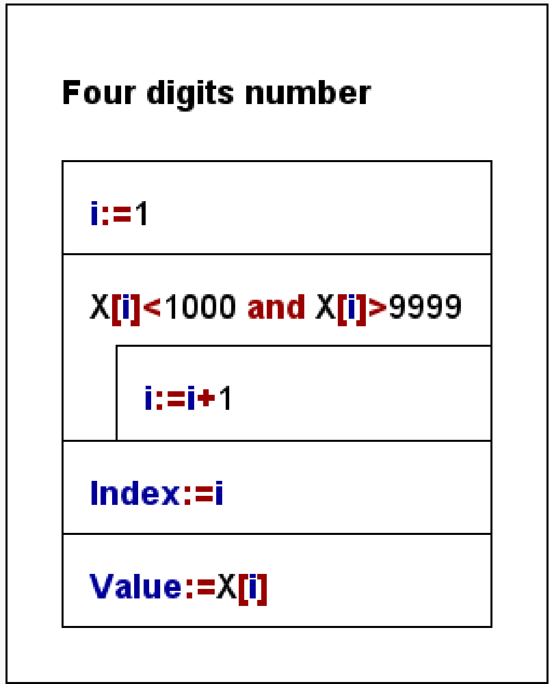
```
Algorithm
i:=1
not A(X[i])
    i:=i+1
ind:=i
val:=X[i]
```

Length(X)	→	N
X[]	→	X[]
ind	→	Index
A(X[i])	→	999 < X[i] <10 000



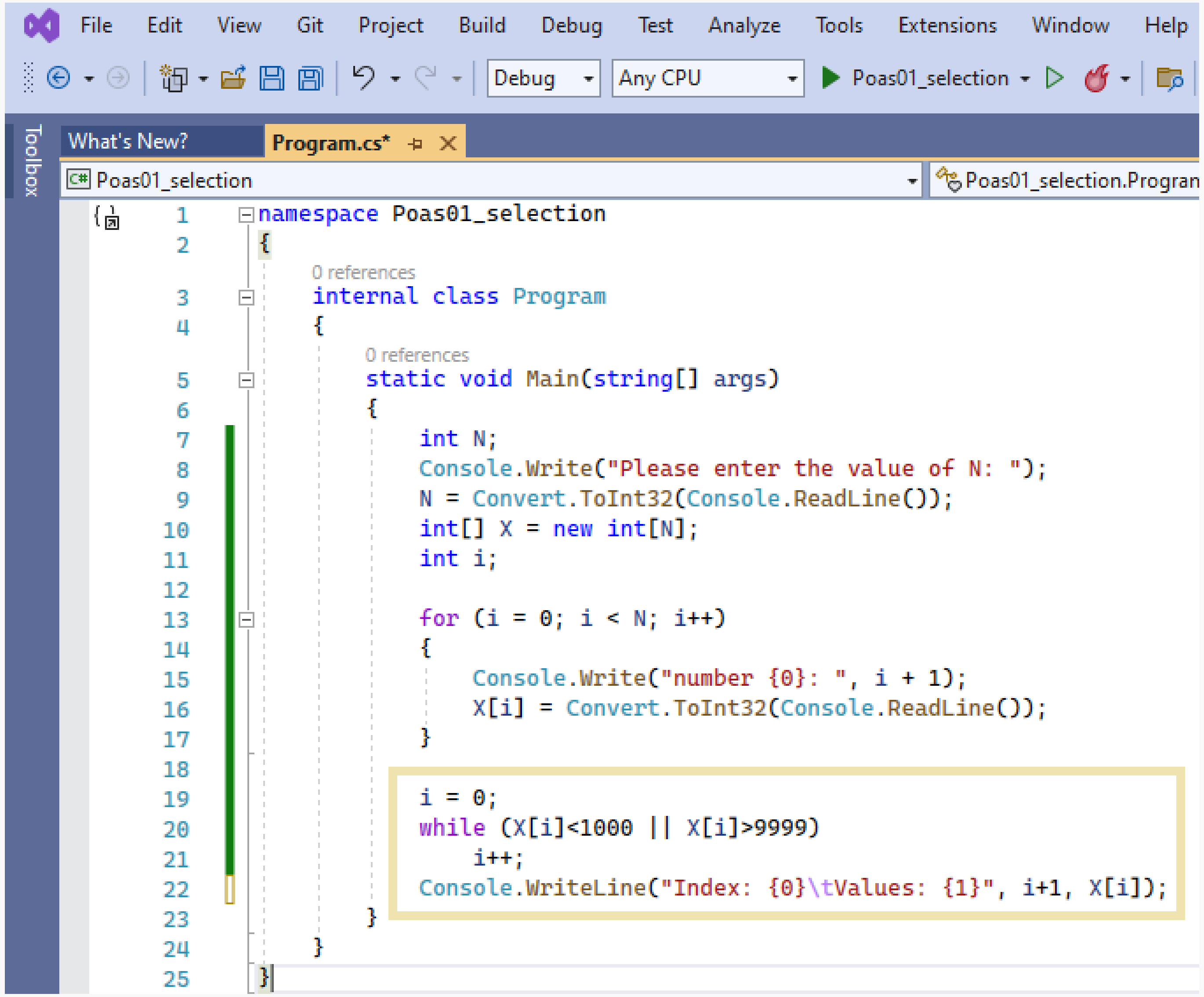
```
Four digits number
i:=1
not (X[i]>999 and X[i]<10000)
    i:=i+1
Index:=i
Value:=X[i]
```

=



```
Four digits number
i:=1
X[i]<1000 and X[i]>9999
    i:=i+1
Index:=i
Value:=X[i]
```

Source code 7 – Selection – Four digits number



```
namespace Poas01_selection
{
    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];
            int i;

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

            i = 0;
            while (X[i]<1000 || X[i]>9999)
                i++;
            Console.WriteLine("Index: {0}\tValues: {1}", i+1, X[i]);
        }
    }
}
```