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:

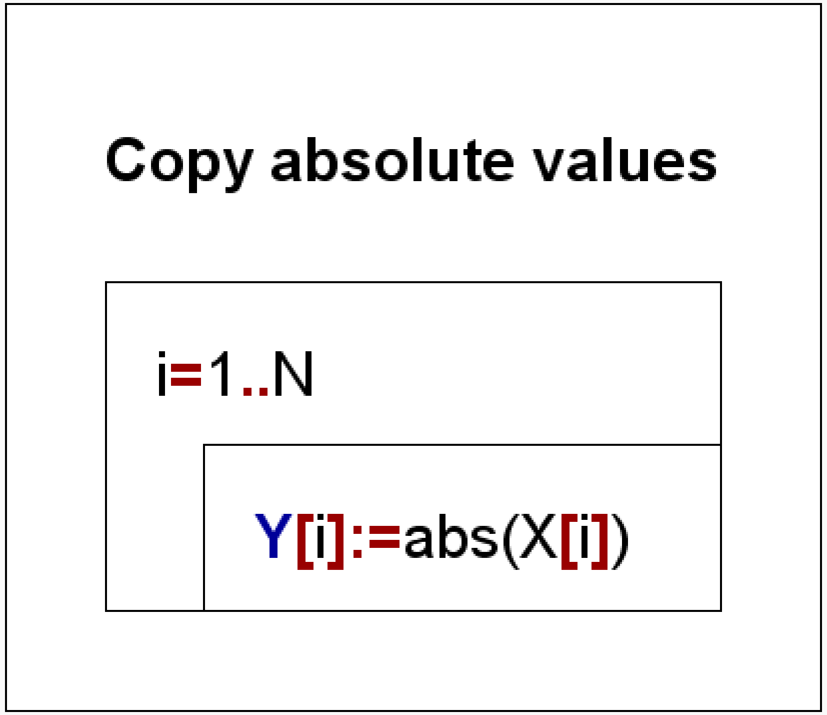
Input: $N \in \mathbb{N}$, $X_{1..N} \in \mathbb{Z}_1^N$

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

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

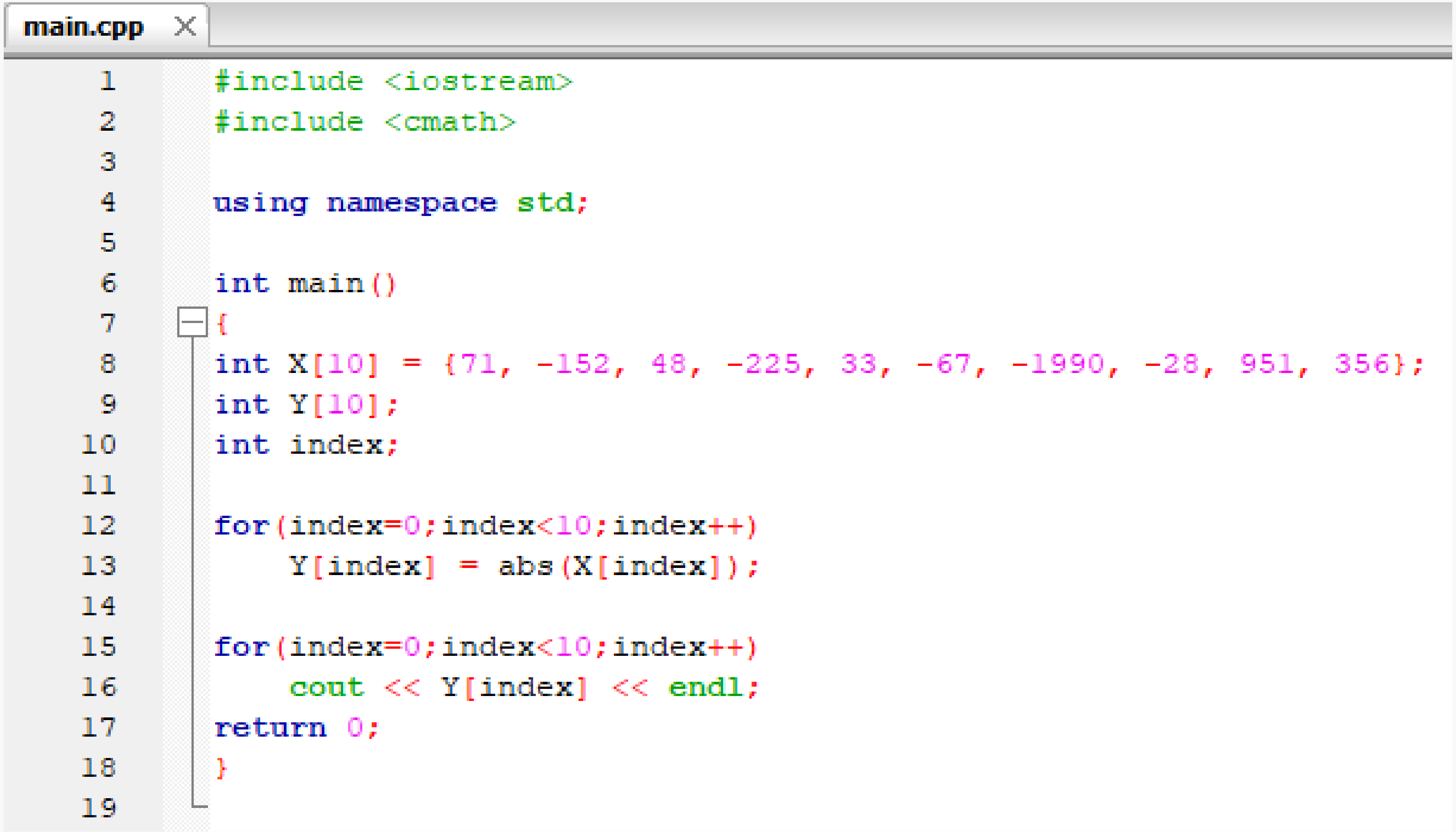
Precondition:

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



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

Source code 1 – Copy – Absolute values



```cpp
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
int X[10] = {71, -152, 48, -225, 33, -67, -1990, -28, 951, 356};
int Y[10];
int index;

for(index=0;index<10;index++)
    Y[index] = abs(X[index]);

for(index=0;index<10;index++)
    cout << Y[index] << endl;
return 0;
}

```

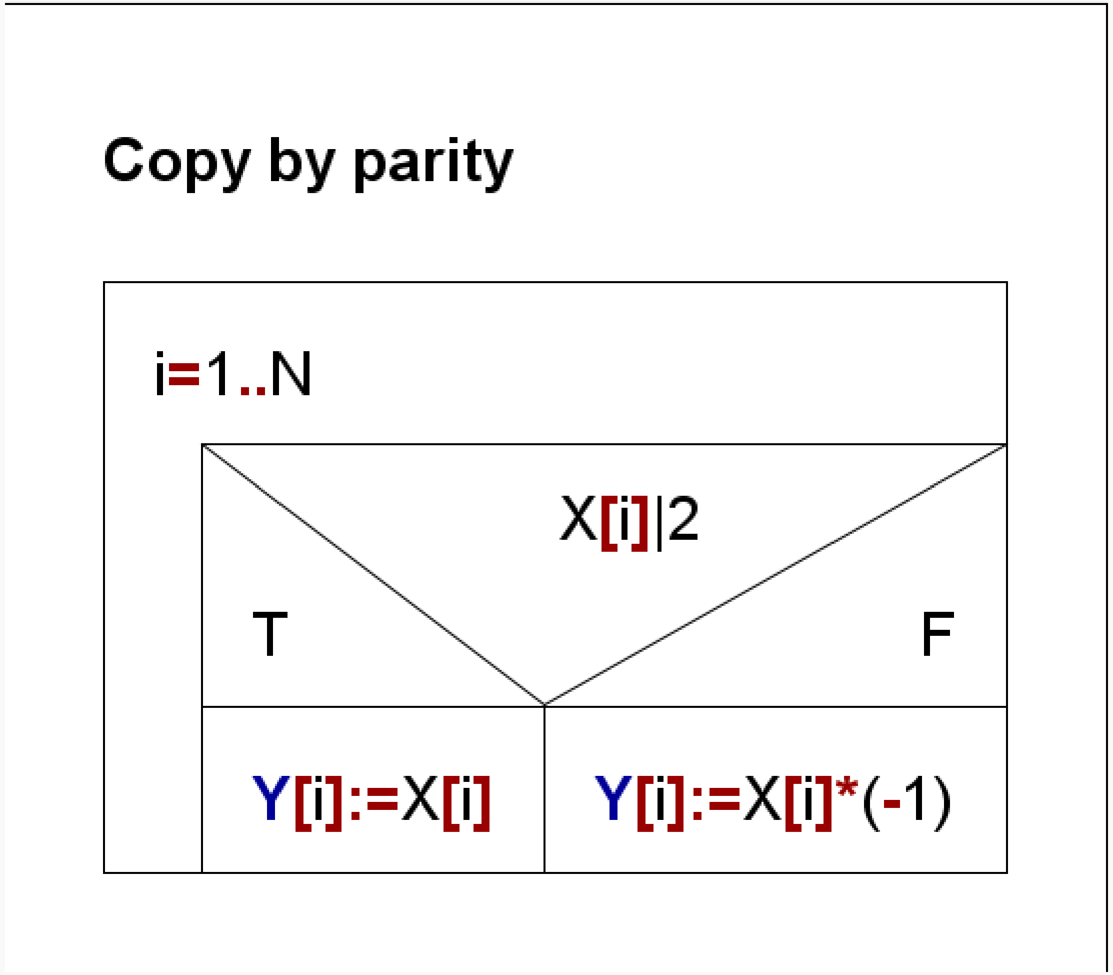
Example 2 – Copy

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

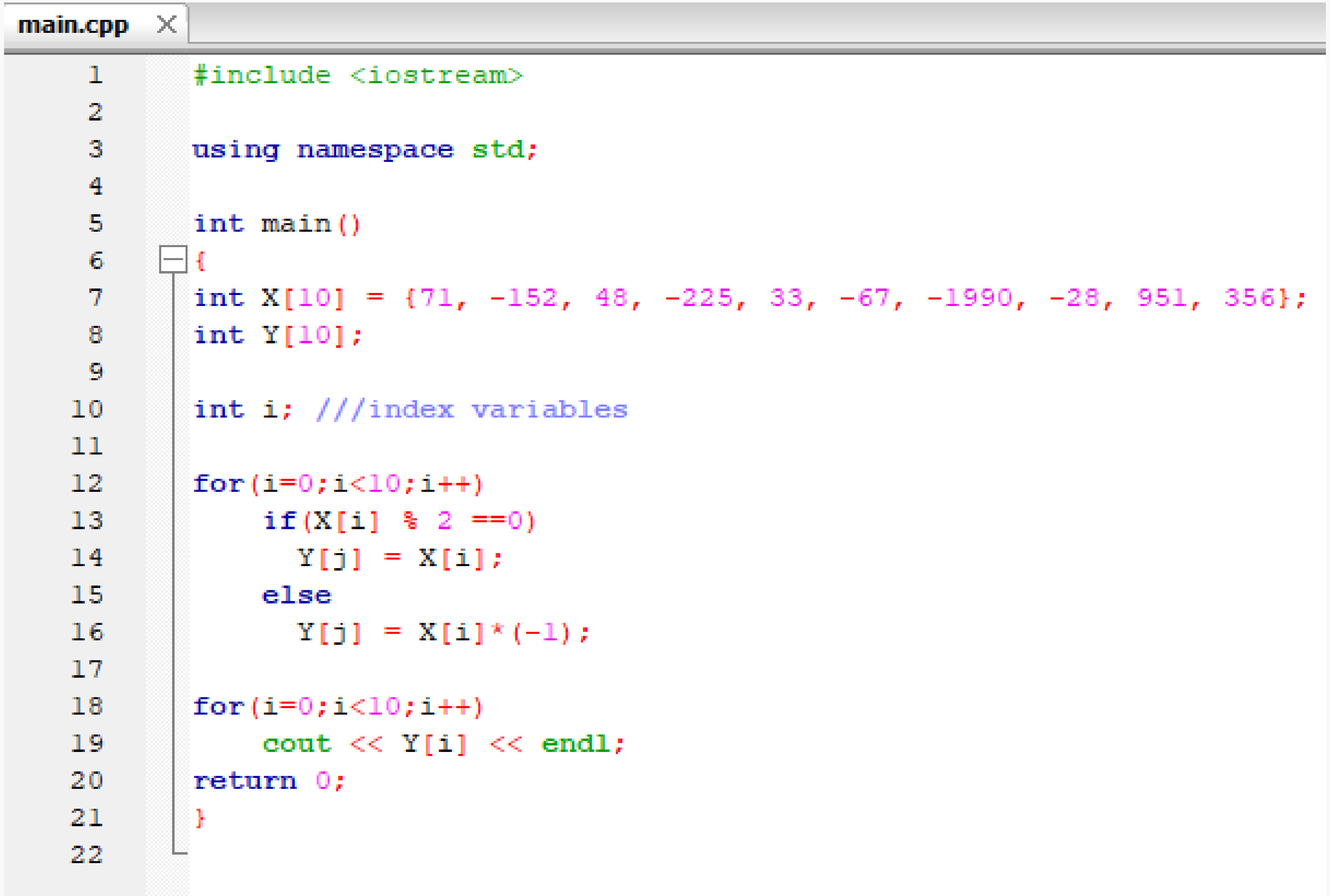
Output: $Y_{1..N} \in \mathbb{Z}^N$
Precondition: -
Postcondition: $\forall$ i ($1 \leq i \leq N$) :
$Y_i = X_i$ and $X_i|2$ and
$Y_i = -X_i$ and not $X_i|2$



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



```cpp
#include <iostream>

using namespace std;

int main()
{
int X[10] = {71, -152, 48, -225, 33, -67, -1990, -28, 951, 356};
int Y[10];

int i; ///index variables

for(i=0;i<10;i++)
    if(X[i] % 2 ==0)
      Y[j] = X[i];
    else
      Y[j] = X[i]*(-1);

for(i=0;i<10;i++)
    cout << Y[i] << endl;
return 0;
}

```

Example 3 – Multiple item selection

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

Specification:

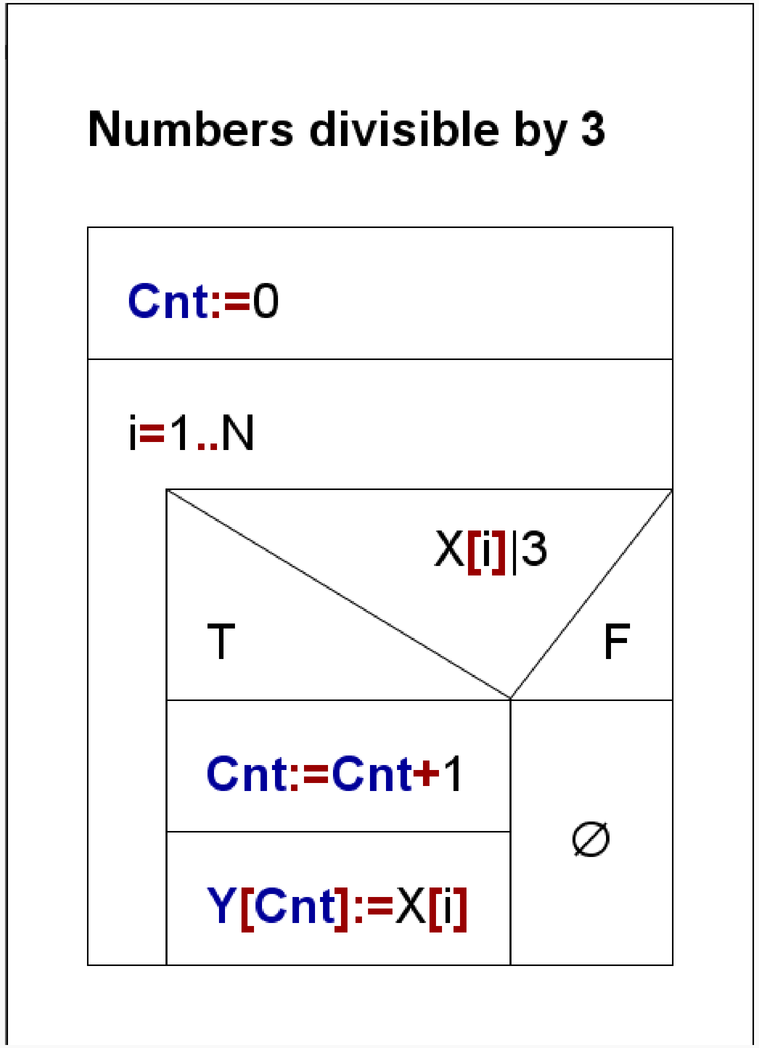
Input: $N \epsilon \mathbb{N}$, $X_{1..N} \epsilon \mathbb{Z}^{N}$ $A: \mathbb{Z} \rightarrow \mathbb{L}, A(x)=x|3$

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

Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ X_i|3}}^{N} 1$ and

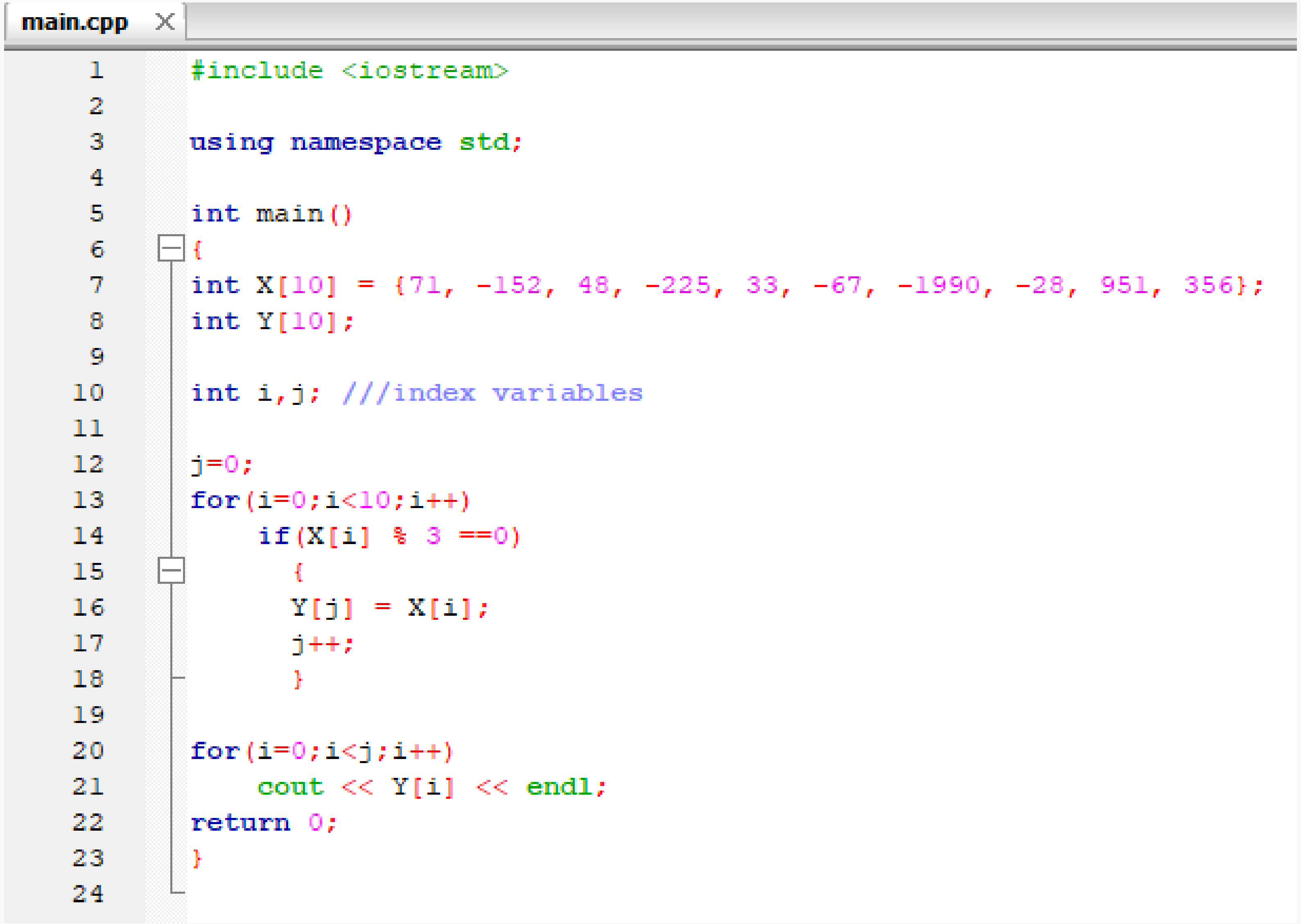
$\forall i\ (1 \leq i \leq Cnt) : Y_i|3$ and $Y \subset X$



Numbers divisible by 3

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

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



```cpp
#include <iostream>

using namespace std;

int main()
{
int X[10] = {71, -152, 48, -225, 33, -67, -1990, -28, 951, 356};
int Y[10];

int i,j; ///index variables

j=0;
for(i=0;i<10;i++)
    if(X[i] % 3 ==0)
      {
      Y[j] = X[i];
      j++;
      }

for(i=0;i<j;i++)
    cout << Y[i] << endl;
return 0;
}

```

Example 4 – Multiple item selection

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

Specification:

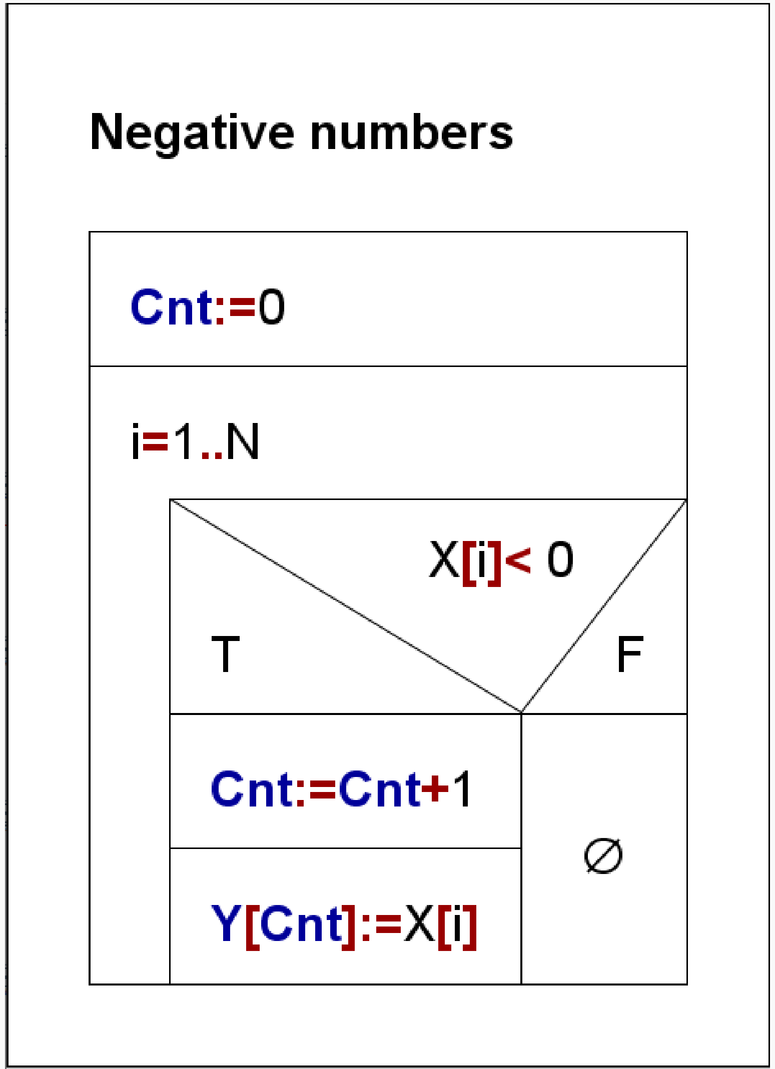
Input: $N \epsilon \mathbb{N}$, $X_{1..N} \epsilon \mathbb{Z}^{N}$ $A: \mathbb{Z} \longrightarrow \mathbb{L}$, $A(x)=x<0$

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

Precondition: –

Postcondition: $Cnt = \sum_{\substack{i=1 \\ X_i<0}}^{N} 1$ and

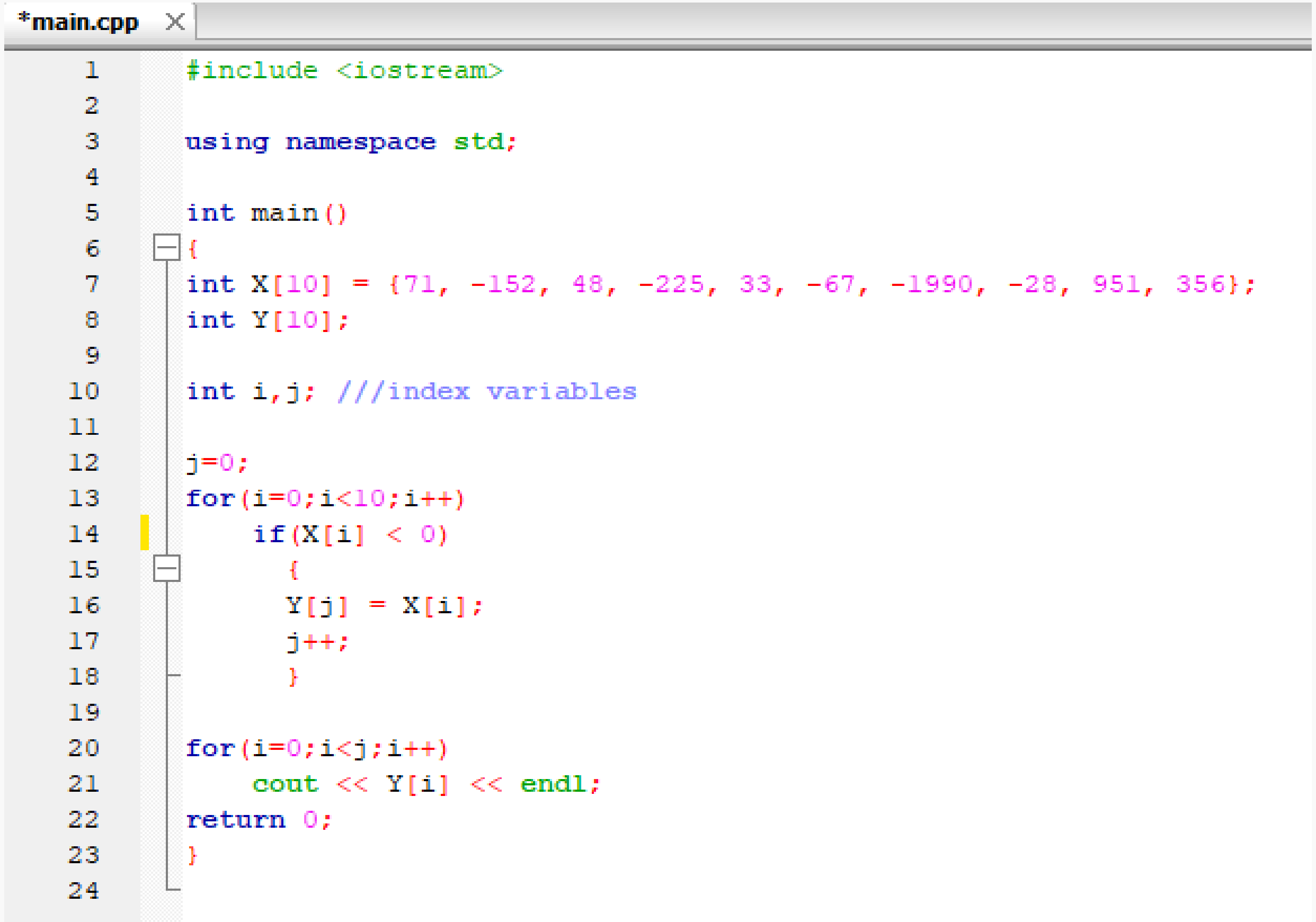
$\forall\ i\ (1 \leq i \leq Cnt) : Y_i < 0$ and $Y \subseteq X$



Negative numbers

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

Example 4 – Multiple item selection – Negative numbers



```cpp
#include <iostream>

using namespace std;

int main()
{
int X[10] = {71, -152, 48, -225, 33, -67, -1990, -28, 951, 356};
int Y[10];

int i,j; ///index variables

j=0;
for(i=0;i<10;i++)
    if(X[i] < 0)
      {
      Y[j] = X[i];
      j++;
      }

for(i=0;i<j;i++)
    cout << Y[i] << endl;
return 0;
}

```

Example 5 – Partitioning

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

Specification:

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

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

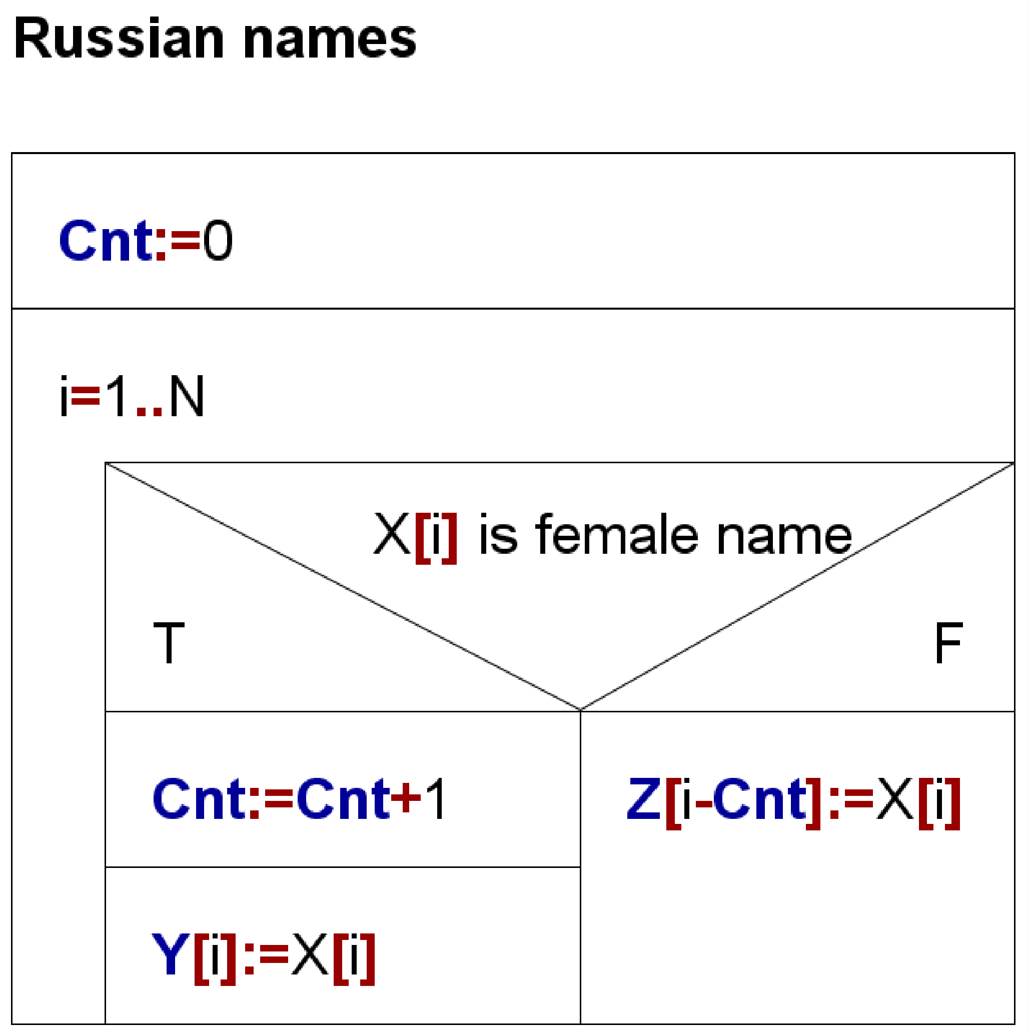
Precondition: –

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

and $\forall$ i ($1 \leq i \leq Cnt$) : Y_i is female name

and $\forall$ i ($1 \leq i \leq N\text{-}Cnt$) : Z_i is not female name

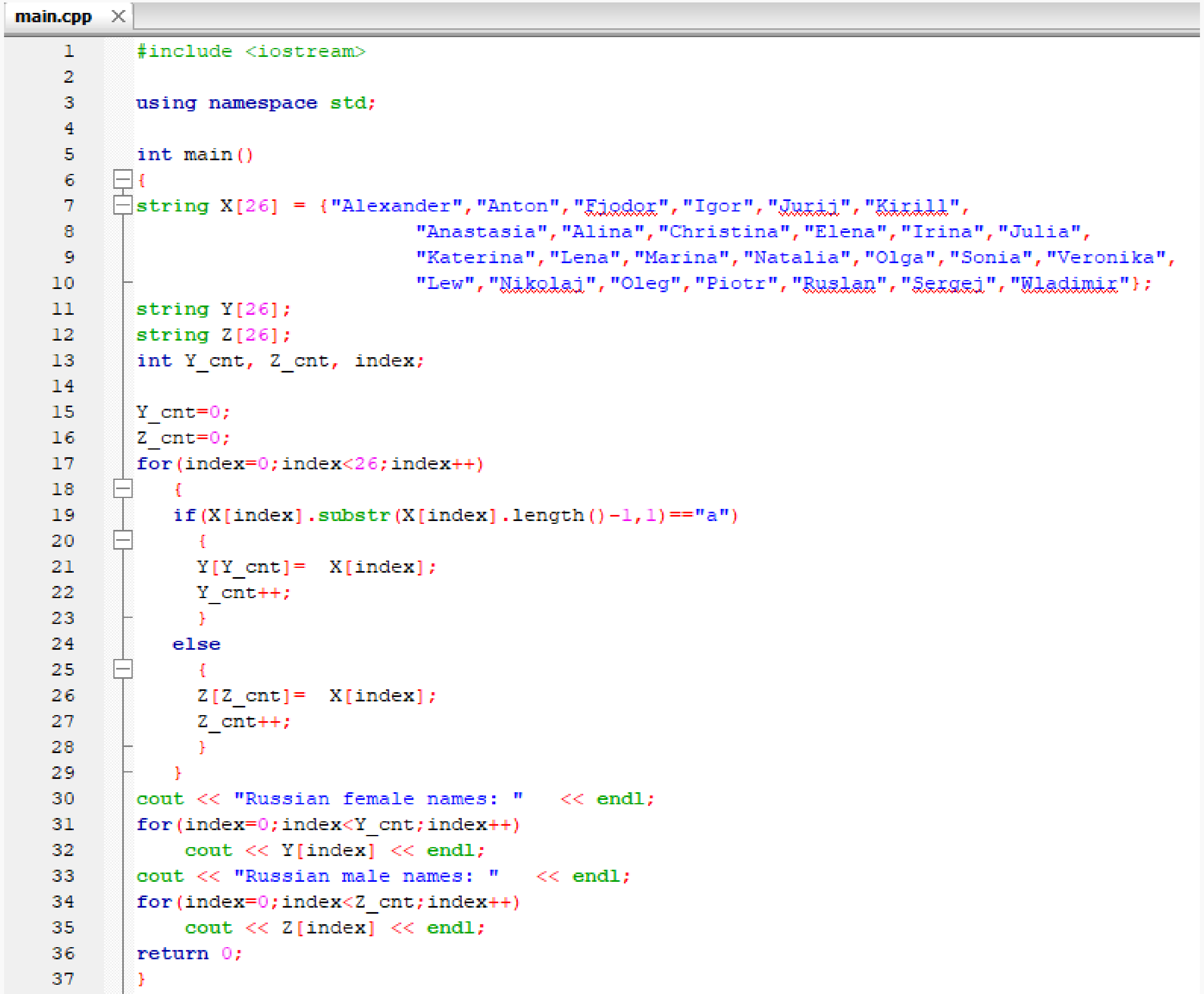
and $Y \subseteq X$ and $Z \subseteq X$



Russian names

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

Example 5 – Partitioning – Russian names



```cpp
#include <iostream>

using namespace std;

int main()
{
string X[26] = {"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[26];
string Z[26];
int Y_cnt, Z_cnt, index;

Y_cnt=0;
Z_cnt=0;
for(index=0;index<26;index++)
   {
   if(X[index].substr(X[index].length()-1,1)=="a")
     {
     Y[Y_cnt]=  X[index];
     Y_cnt++;
     }
   else
     {
     Z[Z_cnt]=  X[index];
     Z_cnt++;
     }
   }
cout << "Russian female names: "   << endl;
for(index=0;index<Y_cnt;index++)
    cout << Y[index] << endl;
cout << "Russian male names: "   << endl;
for(index=0;index<Z_cnt;index++)
    cout << Z[index] << endl;
return 0;
}
```

Example 6 – Intersection

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

Specification:

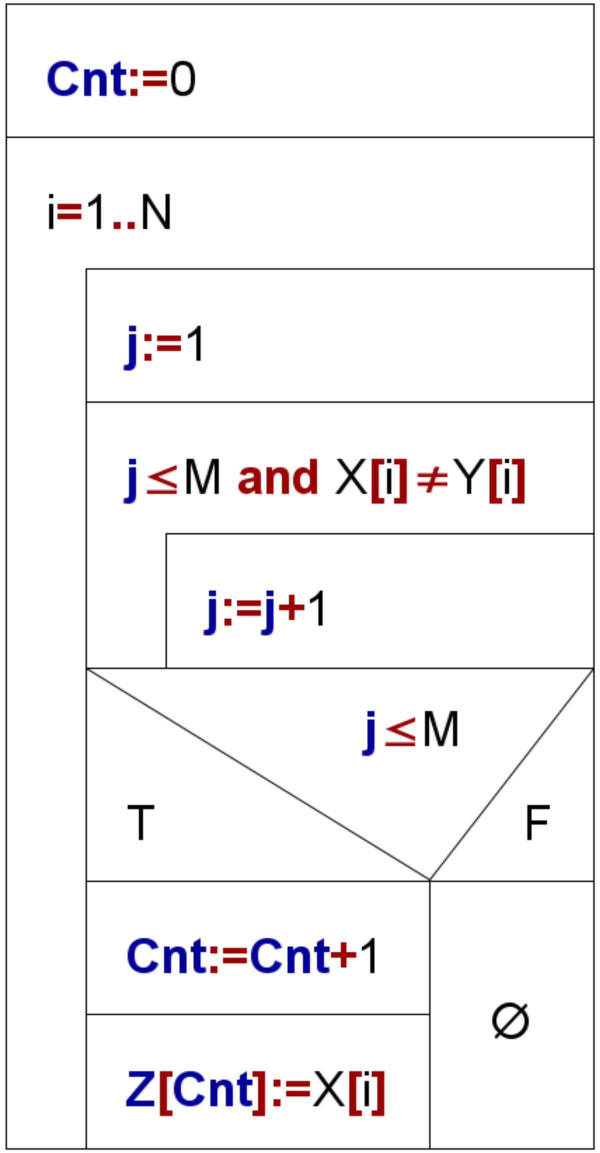
Input: N, M $\in \mathbb{N}$, $X_{1..N} \in \mathbb{T}^{N}$, $Y_{1..M} \in \mathbb{T}^{M}$

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

Precondition: IsSet(X) and IsSet(Y)

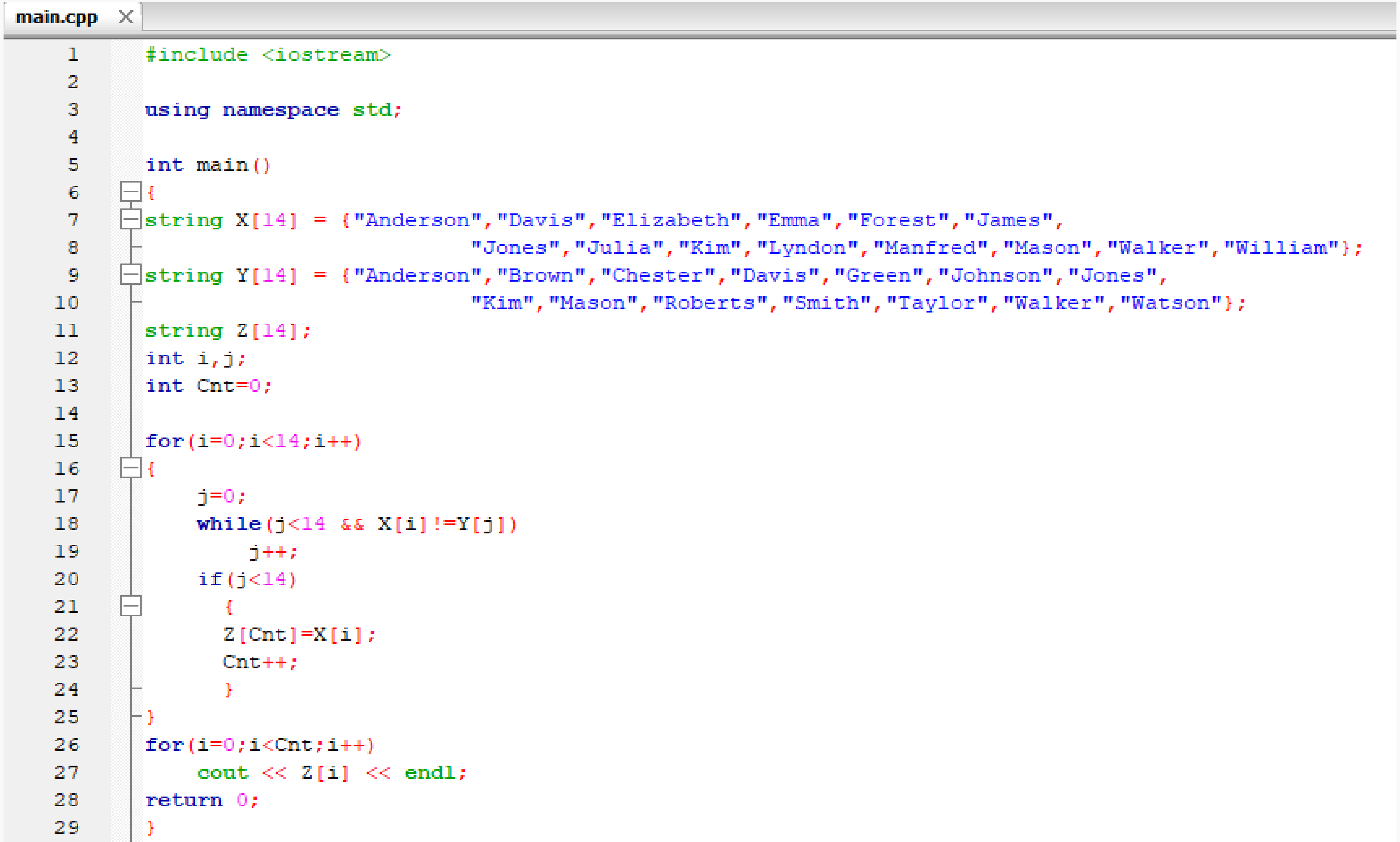
Postcondition: $Cnt = \sum_{\substack{i=1 \\ X_i \in Y}}^{N} 1$

and $\forall$ i ($1 \leq i \leq Cnt$) : $Z_i \in Y$ and IsSet(Z)



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

Example 6 – Intersection – English names



```cpp
#include <iostream>

using namespace std;

int main()
{
string X[14] = {"Anderson","Davis","Elizabeth","Emma","Forest","James",
                         "Jones","Julia","Kim","Lyndon","Manfred","Mason","Walker","William"};
string Y[14] = {"Anderson","Brown","Chester","Davis","Green","Johnson","Jones",
                         "Kim","Mason","Roberts","Smith","Taylor","Walker","Watson"};
string Z[14];
int i,j;
int Cnt=0;

for(i=0;i<14;i++)
{
    j=0;
    while(j<14 && X[i]!=Y[j])
        j++;
    if(j<14)
      {
      Z[Cnt]=X[i];
      Cnt++;
      }
}
for(i=0;i<Cnt;i++)
    cout << Z[i] << endl;
return 0;
}
```

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

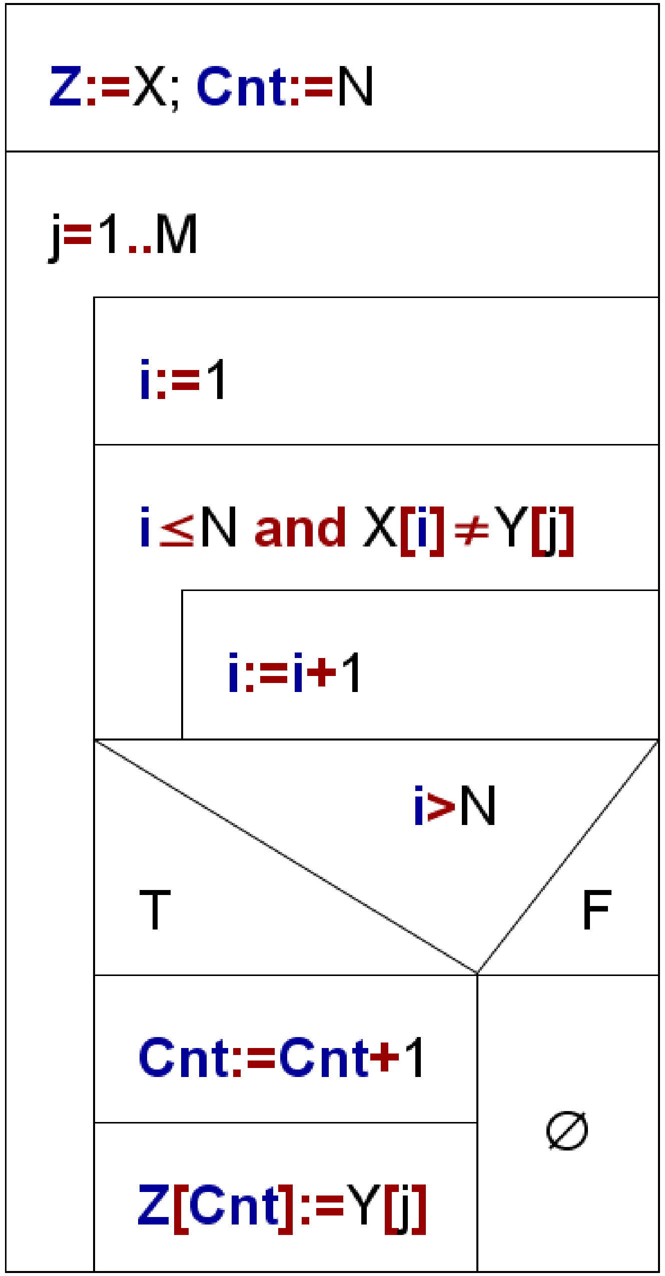
Output: Cnt $\in \mathbb{N}$, $Z_{1..N+M} \in \mathbb{T}^{N+M}$

Precondition: IsSet(X) and IsSet(Y)

Postcondition: $\text{Cnt} = \text{N} + \sum_{\substack{j=1 \\ Y_j \notin X}}^{N} 1$

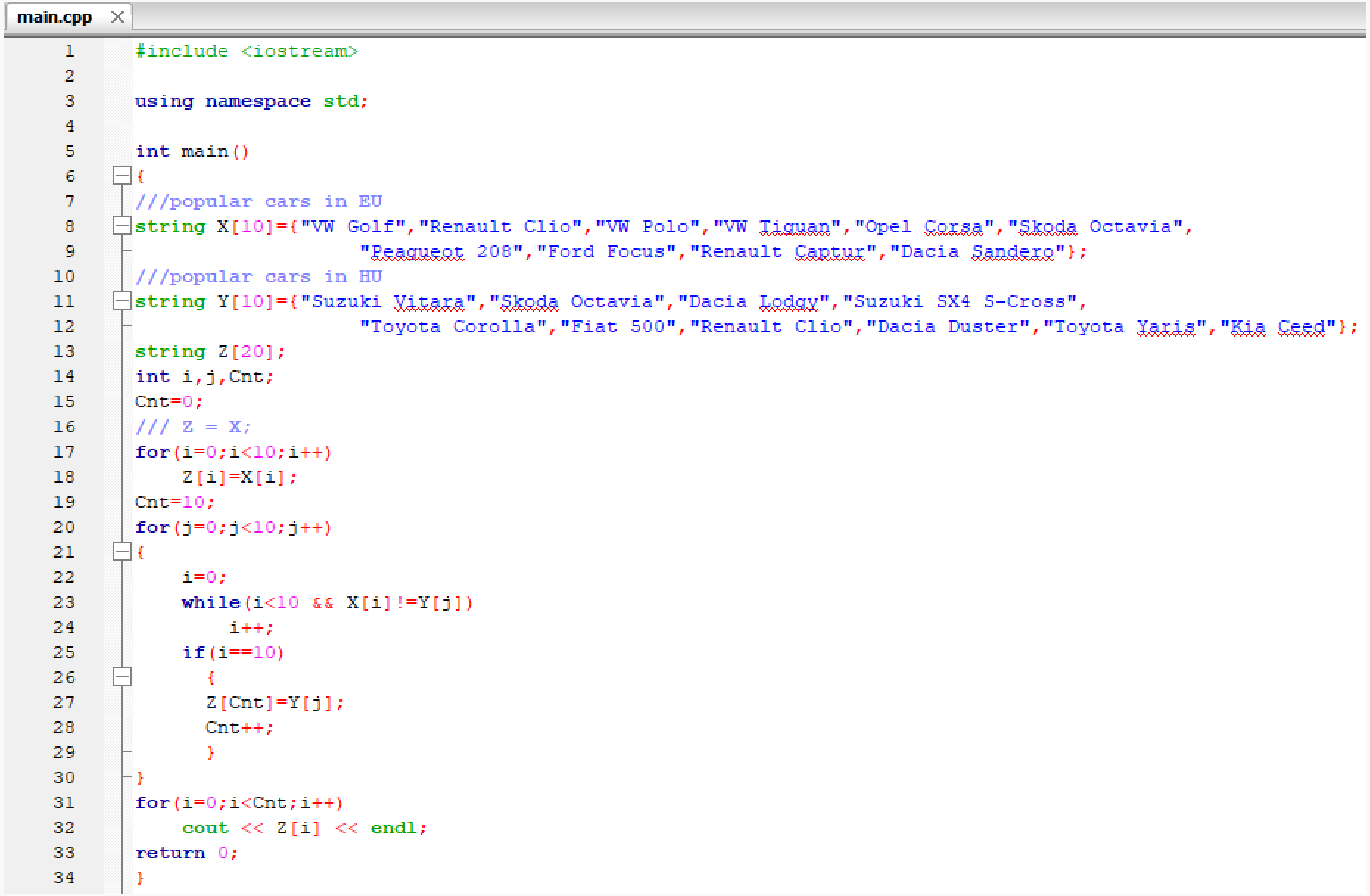
and $\forall$ i ($1 \leq i \leq$ Cnt) : $Z_i \in X$ or $Z_i \in Y$ and IsSet(Z)

Cars



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

Example 7 – Union – Popular cars



```cpp
#include <iostream>

using namespace std;

int main()
{
///popular cars in EU
string X[10]={"VW Golf","Renault Clio","VW Polo","VW Tiguan","Opel Corsa","Skoda Octavia",
                   "Peagueot 208","Ford Focus","Renault Captur","Dacia Sandero"};
///popular cars in HU
string Y[10]={"Suzuki Vitara","Skoda Octavia","Dacia Lodgy","Suzuki SX4 S-Cross",
                   "Toyota Corolla","Fiat 500","Renault Clio","Dacia Duster","Toyota Yaris","Kia Ceed"};
string Z[20];
int i,j,Cnt;
Cnt=0;
/// Z = X;
for(i=0;i<10;i++)
    Z[i]=X[i];
Cnt=10;
for(j=0;j<10;j++)
{
    i=0;
    while(i<10 && X[i]!=Y[j])
        i++;
    if(i==10)
      {
      Z[Cnt]=Y[j];
      Cnt++;
      }
}
for(i=0;i<Cnt;i++)
    cout << Z[i] << endl;
return 0;
}
```