



## Ejemplo de Creación de Tabla de símbolos

Diseño de Compiladores

Dr. Víctor de la Cueva

[vcueva@itesm.mx](mailto:vcueva@itesm.mx)

## Código

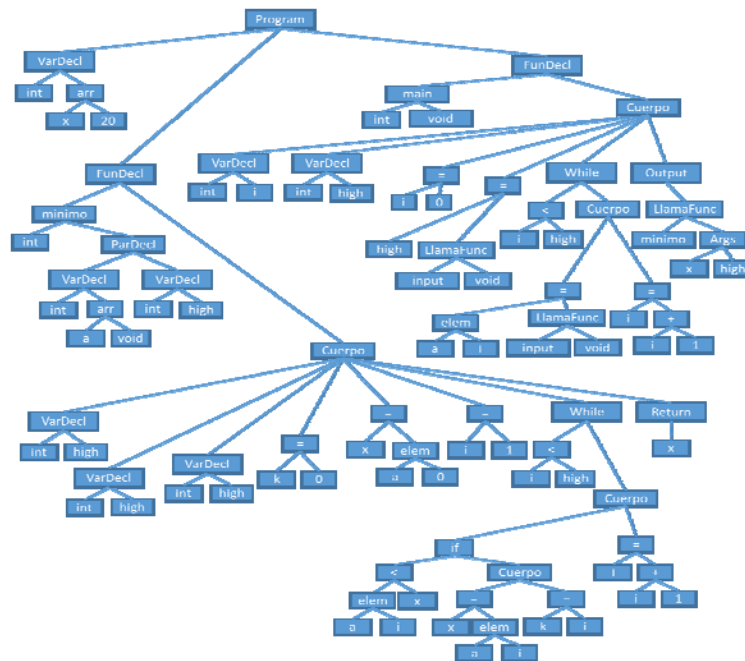
```
/* Un programa para encontrar el mínimo de
   un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while (i < high)
  { if (a[i] < x)
    { x = a[i];
      k = i; }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while (i < high)
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}
```

## AST



```

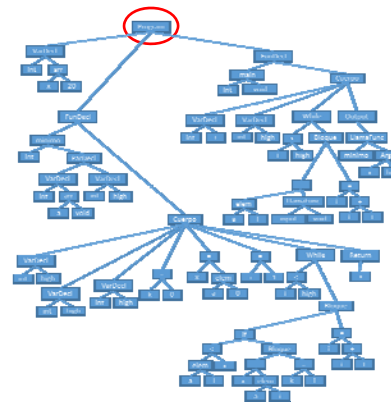
/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if ( a[i] < x )
    { x = a[i];
      k = i; }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}

```



Inicializar Scope a 0  
Ver el **current scope** pointer.

Scope 0

Tabla de Símbolos

← Current Scope

```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if (a[i] < x)
    { x = a[i];
      k = i; }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}

```

Agregar la variable arreglo x al **current scope**.

x   int   arr   10 <b>Scope 0</b>
--------------------------------------

← Current Scope

Tabla de Símbolos

```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if (a[i] < x)
    { x = a[i];
      k = i; }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}

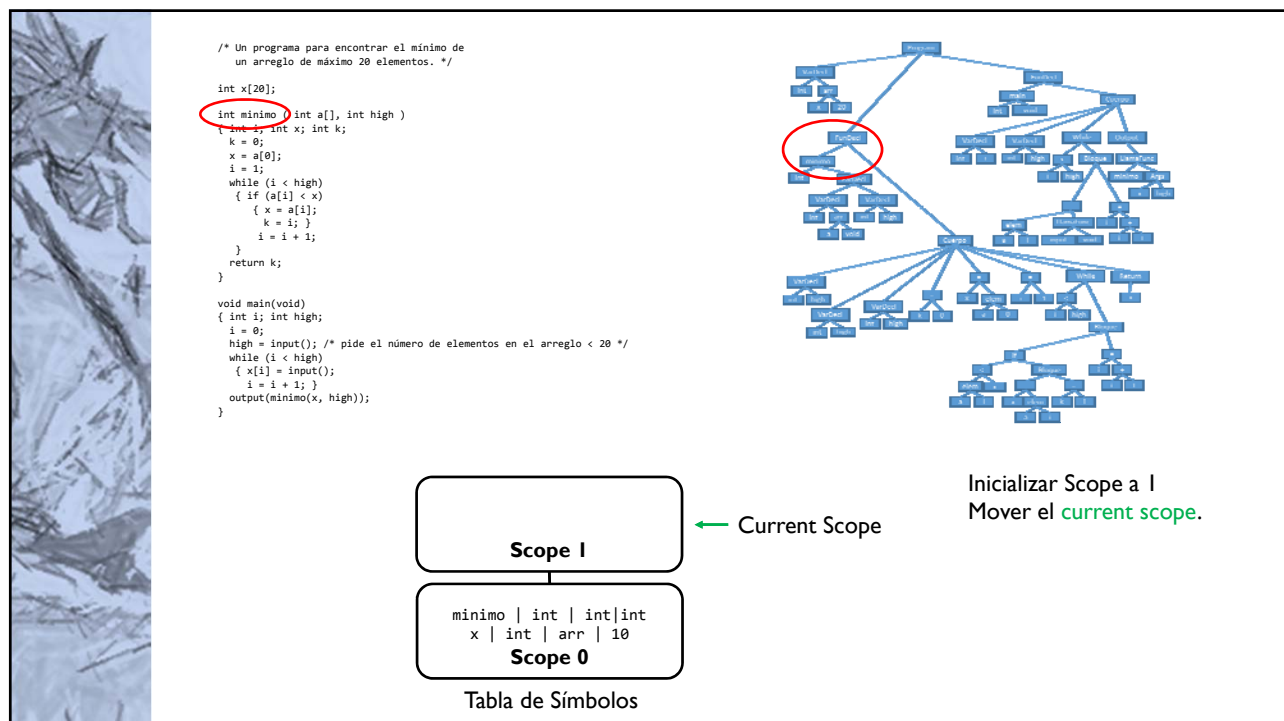
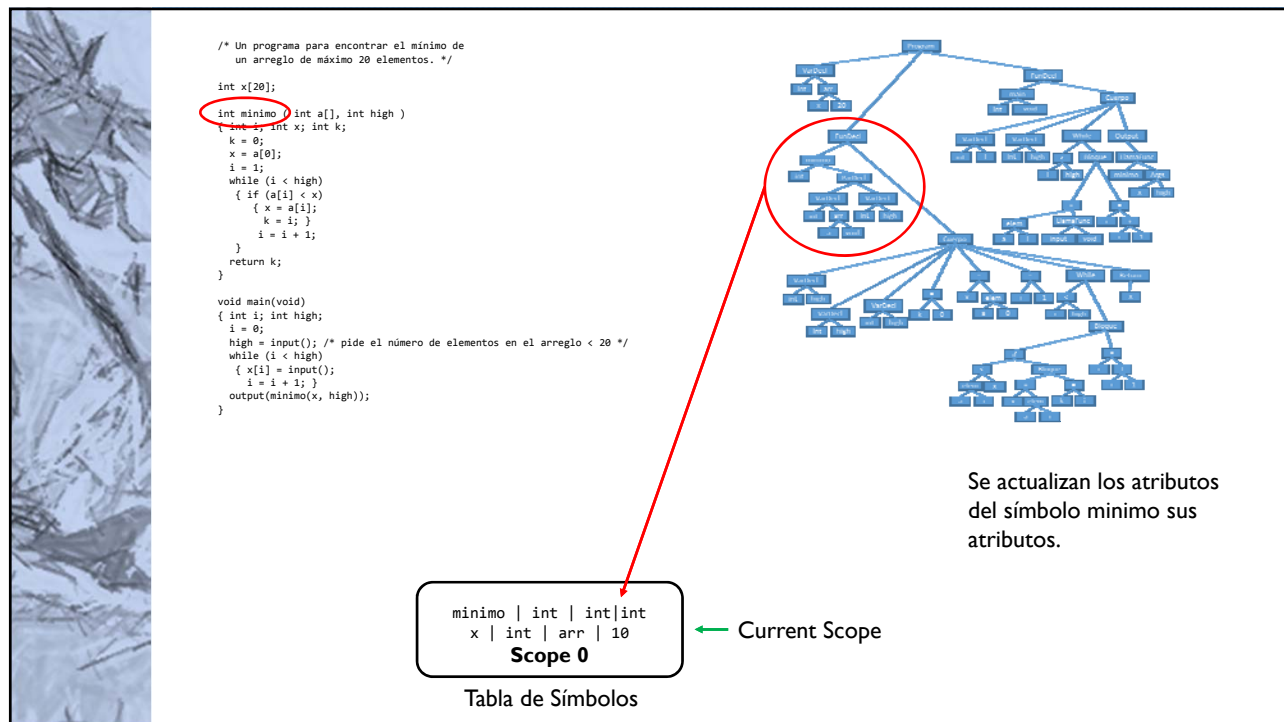
```

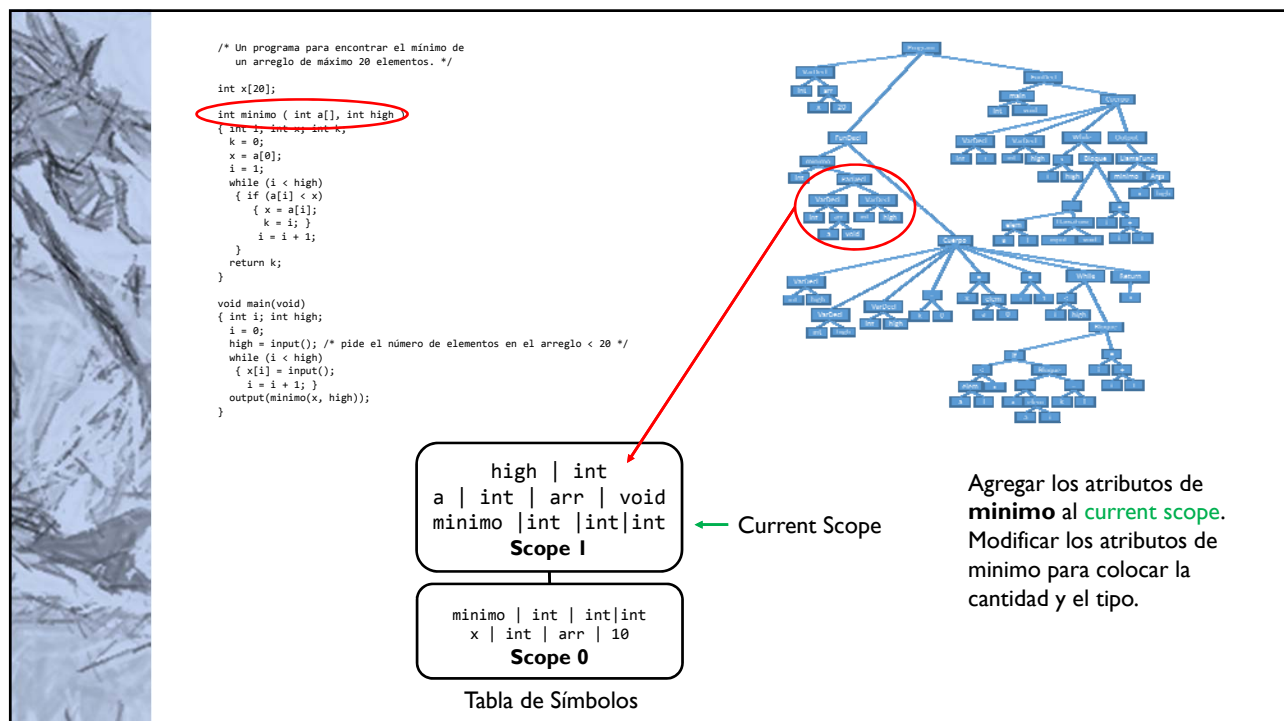
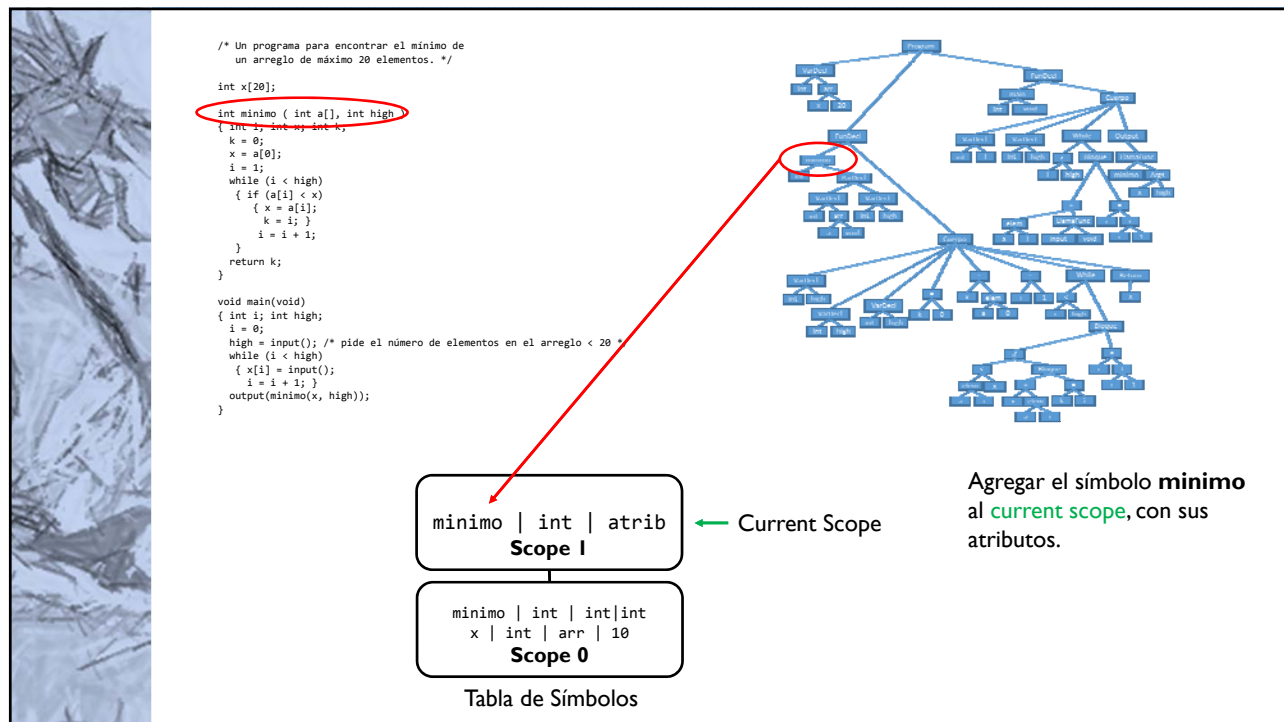
Se agrega el símbolo minimo al **current scope** con su tipo y sus atributos.

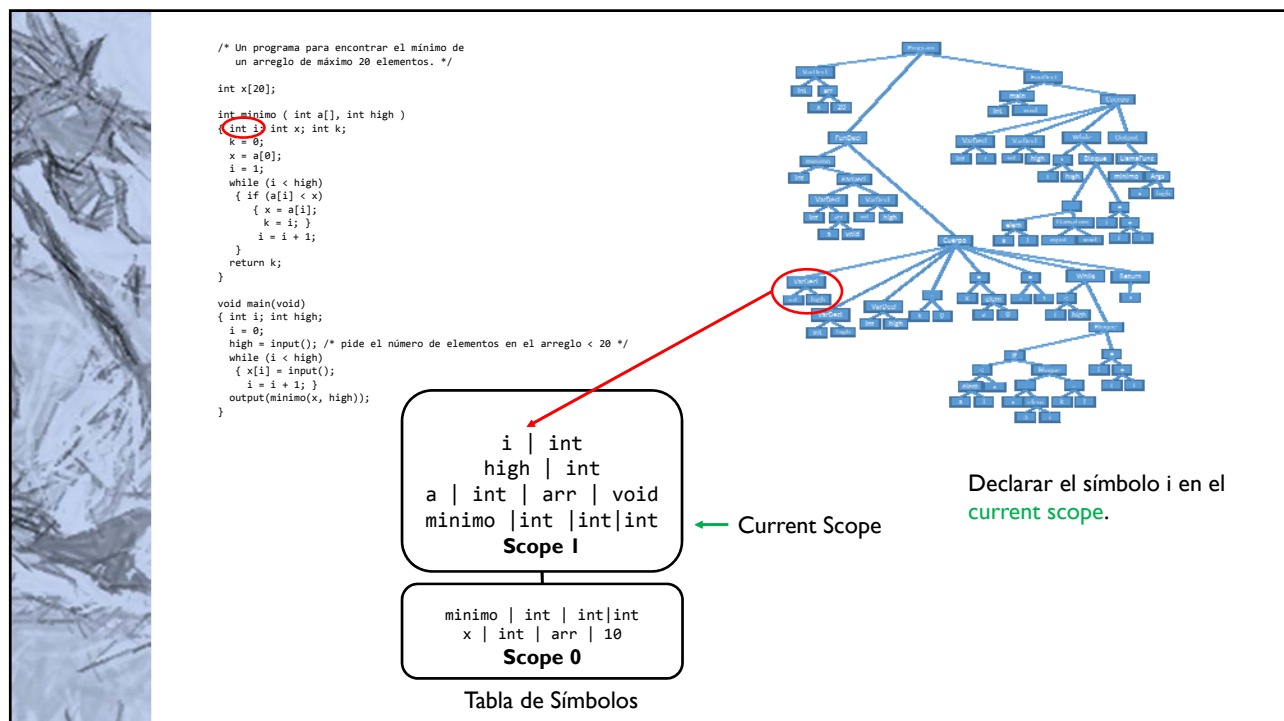
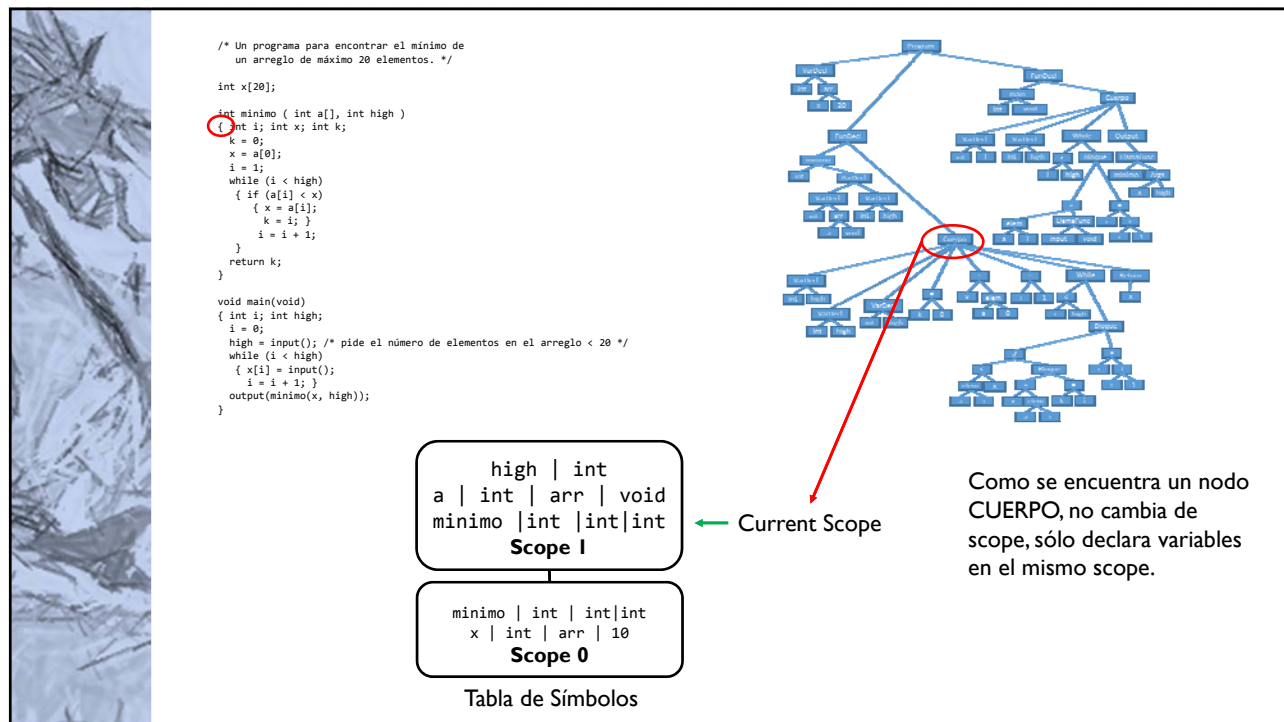
minimo   int   atrib x   int   arr   10 <b>Scope 0</b>
--

← Current Scope

Tabla de Símbolos









```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if ( a[i] < x )
    { x = a[i];
      k = i;
    }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1;
  }
  output(minimo(x, high));
}

```

x	int
i	int
high	int
a	int   arr   void
minimo	int   int   int
<b>Scope I</b>	

minimo	int   int   int   int
x	int   arr   10
<b>Scope 0</b>	

Tabla de Símbolos

← Current Scope

Declarar el símbolo x en el current scope.

```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if ( a[i] < x )
    { x = a[i];
      k = i;
    }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1;
  }
  output(minimo(x, high));
}

```

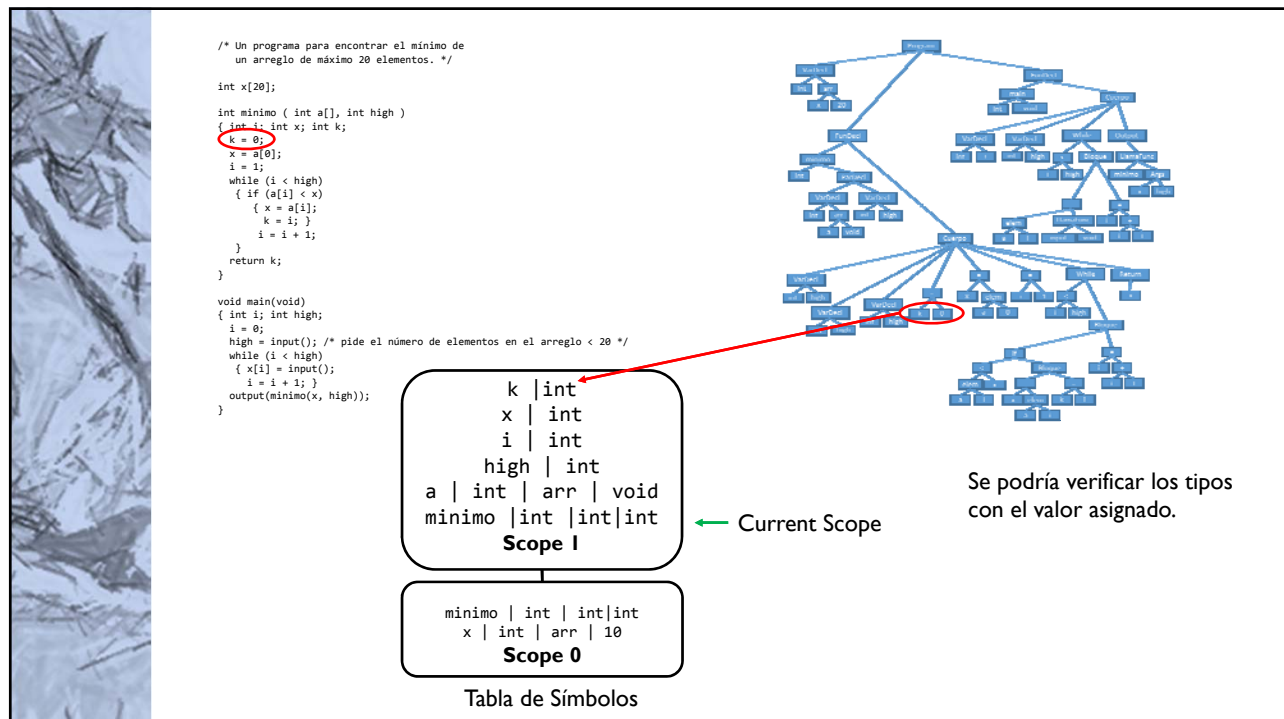
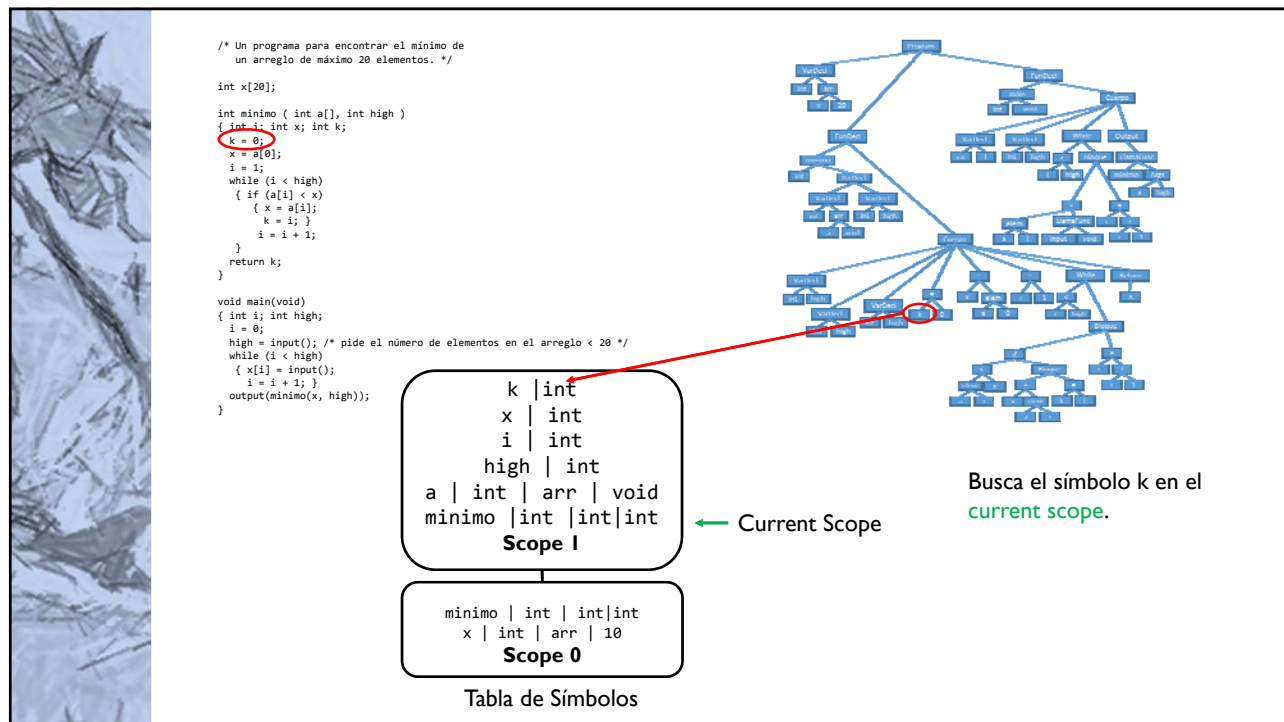
k	int
x	int
i	int
high	int
a	int   arr   void
minimo	int   int   int
<b>Scope I</b>	

minimo	int   int   int   int
x	int   arr   10
<b>Scope 0</b>	

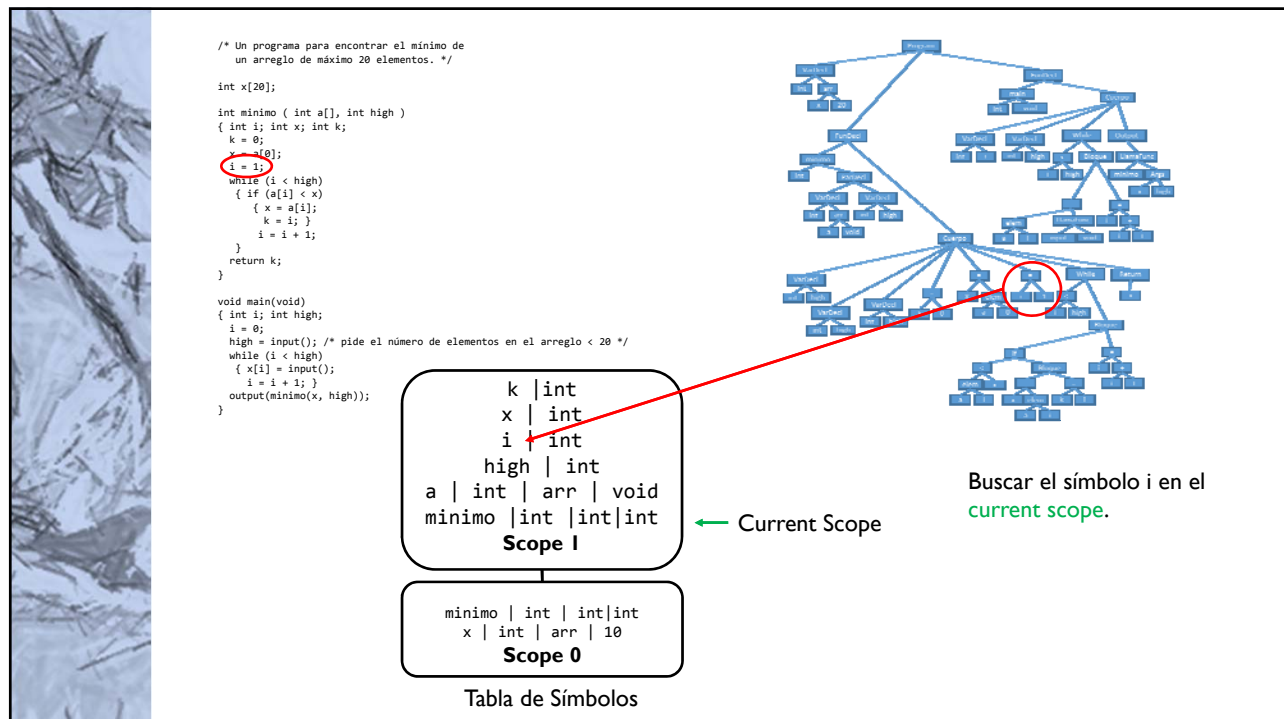
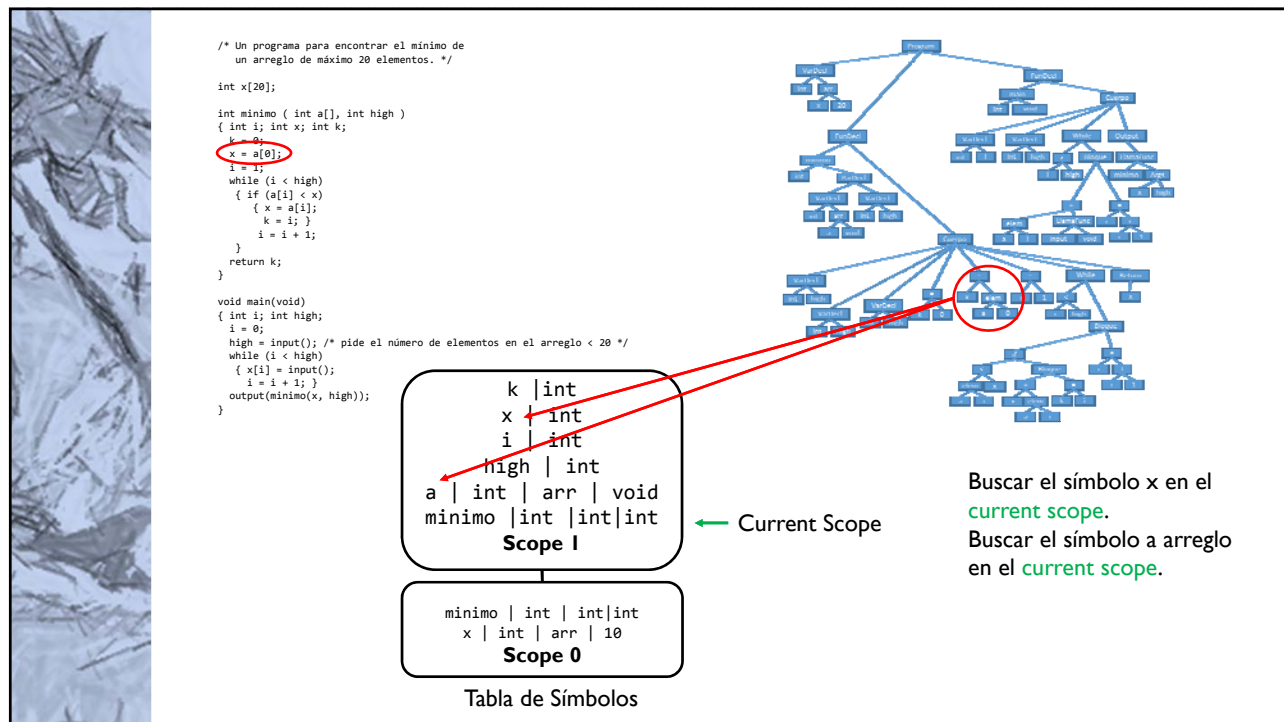
Tabla de Símbolos

← Current Scope

Declarar el símbolo k en el current scope.







```

    k | int
    x | int
    i | int
    high | int
a | int | arr | void
minimo | int | int | int
Scope 1

```

```

minimo | int | int | int
x | int | arr | 10
Scope 0

```

[illegible]

← Current Scope

The diagram illustrates the scope resolution process for the variable 'high'. It consists of two rounded rectangular boxes representing different scopes. The top box is labeled 'Scope 1' and contains the following declarations: 'k | int', 'x | int', 'i | int', 'high | int', 'a | int | arr | void', and 'minimo | int | int | int'. The bottom box is labeled 'Scope 0' and contains: 'minimo | int | int | int', 'x | int | arr | 10'. A red arrow points from the 'high' declaration in Scope 1 down to the 'high' declaration in Scope 0, indicating that the variable in Scope 0 is the one being resolved.

```
k | int
x | int
i | int
high | int
a | int | arr | void
minimo | int | int | int
Scope 1
```

```
minimo | int | int | int
x | int | arr | 10
Scope 0
```

[illegible]

← Current Scope

```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { f ( a[i] < x )
    { x = a[i];
      k = i;
    }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1;
  }
  output(minimo(x, high));
}

```

k		int
x		int
i		int
high		int
a		int   arr   void
minimo		int   int   int
<b>Scope 1</b>		

minimo		int   int   int
x		int   arr   10
<b>Scope 0</b>		

Tabla de Símbolos

← Current Scope

Se encuentra el nodo CUERPO del while y se sabe que no se debe crear un nuevo scope.

```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if ( a[i] < x )
    { x = a[i];
      k = i;
    }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1;
  }
  output(minimo(x, high));
}

```

k		int
x		int
i		int
high		int
a		int   arr   void
minimo		int   int   int
<b>Scope 1</b>		

minimo		int   int   int
x		int   arr   10
<b>Scope 0</b>		

Tabla de Símbolos

← Current Scope

La primera instrucción del cuerpo es un nodo IF y se sabe que su hijo izquierdo es la condición.

```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while (i < high)
  { if (a[i] < x)
    { x = a[i];
      k = i;
    }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while (i < high)
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}

```

Buscar el símbolo arreglo a en el **current scope**.  
Buscar el símbolo x en el **current scope**.

k   int
x   int
i   int
high   int
a   int   arr   void
minimo   int   int   int
<b>Scope 1</b>

minimo   int   int   int
x   int   arr   10
<b>Scope 0</b>

Tabla de Símbolos

```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while (i < high)
  { if (a[i] < x)
    { x = a[i];
      k = i;
    }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while (i < high)
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}

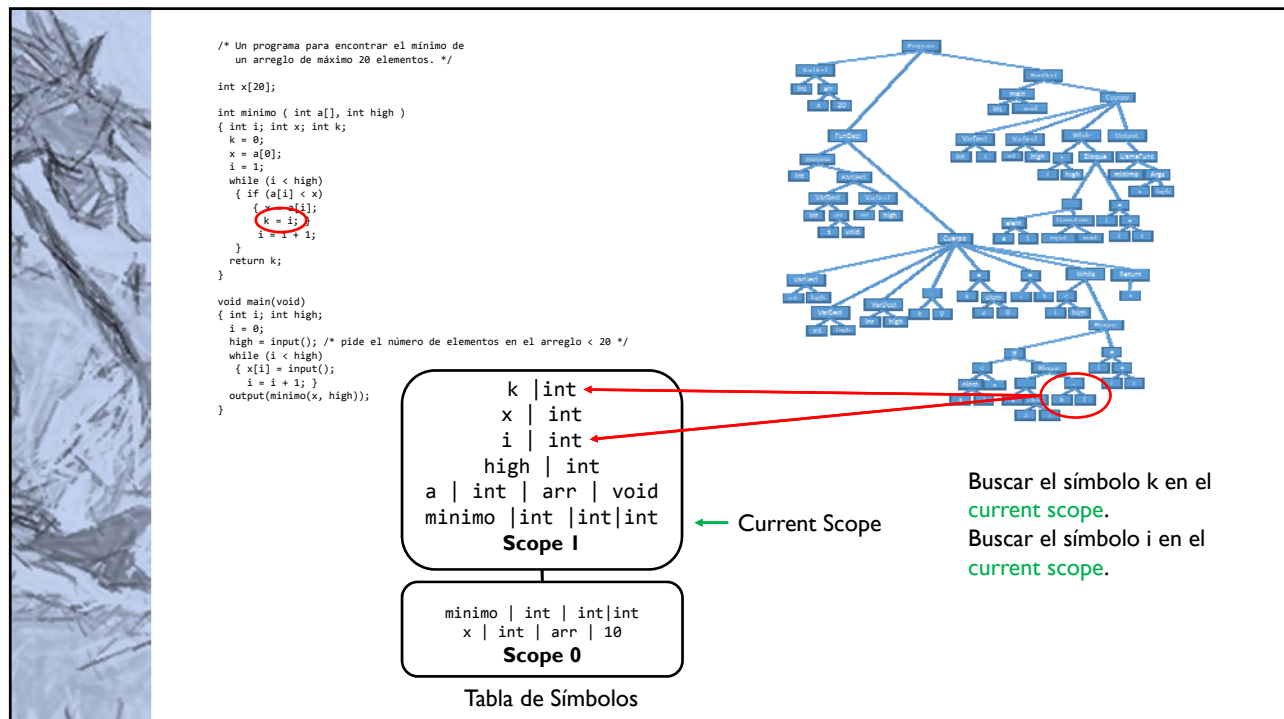
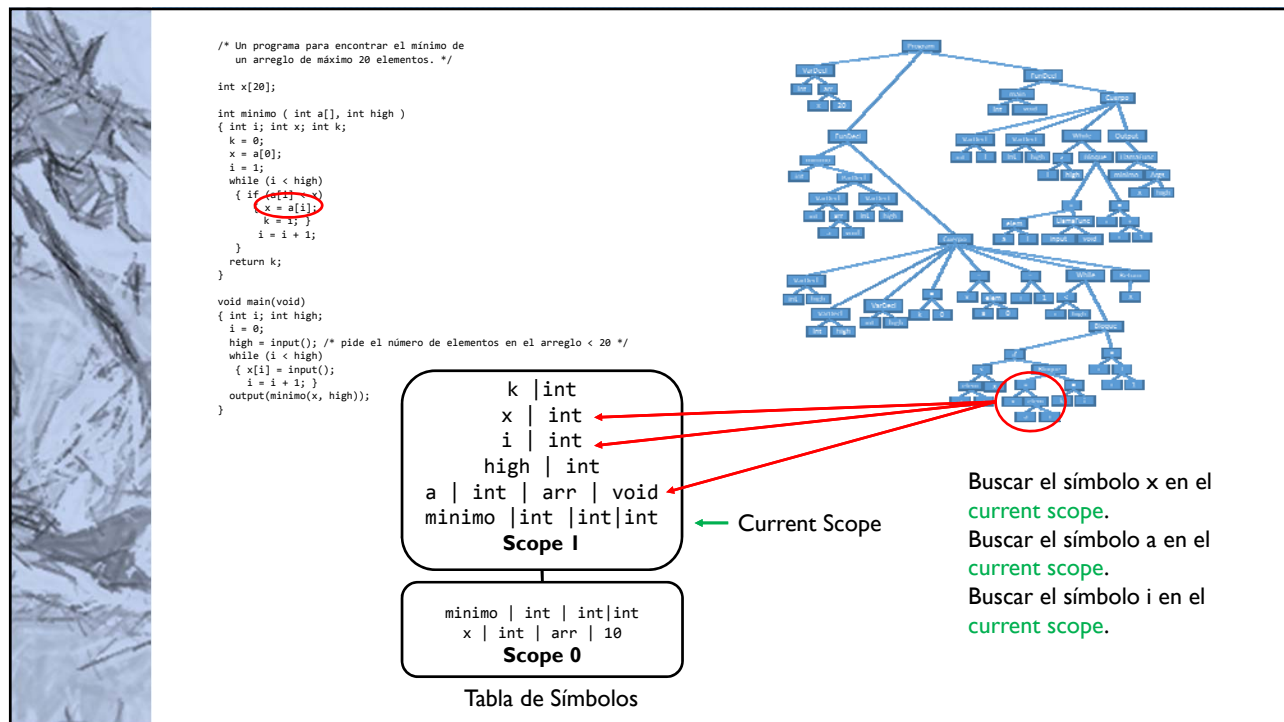
```

Encuentra el nodo CUERPO del if y se sabe que no se debe crear un nuevo scope.

k   int
x   int
i   int
high   int
a   int   arr   void
minimo   int   int   int
<b>Scope 1</b>

minimo   int   int   int
x   int   arr   10
<b>Scope 0</b>

Tabla de Símbolos



```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if (a[i] < x)
    { x = a[i];
      i = i + 1;
    }
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}

```

k		int
x		int
i		int
high		int
a		int   arr   void
minimo		int   int   int
<b>Scope I</b>		

minimo		int   int   int
x		int   arr   10
<b>Scope 0</b>		

Tabla de Símbolos

← Current Scope

Buscar el símbolo i en el **current scope**.  
 Buscar el símbolo i en el **current scope**.  
 Dos veces porque está de los dos lados.

```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if (a[i] < x)
    { x = a[i];
      k = i;
      i = i + 1;
    }
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}

```

k		int
x		int
i		int
high		int
a		int   arr   void
minimo		int   int   int
<b>Scope I</b>		

minimo		int   int   int
x		int   arr   10
<b>Scope 0</b>		

Tabla de Símbolos

← Current Scope

Buscar el símbolo k en el **current scope**.



```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if ( a[i] < x )
    { x = a[i];
      k = i;
    }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}

```

Cerrar el scope I y regresar el **current scope** a 0.

k   int
x   int
i   int
high   int
a   int   arr   void
minimo   int   int   int
<b>Scope I</b>

minimo   int   int   int
x   int   arr   10
<b>Scope 0</b>

← Current Scope

Tabla de Símbolos

```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if ( a[i] < x )
    { x = a[i];
      k = i;
    }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1; }
  output(minimo(x, high));
}

```

Se encuentra un nodo de declaración de función y se agrega el símbolo main al **current scope**, con su tipo y argumentos.

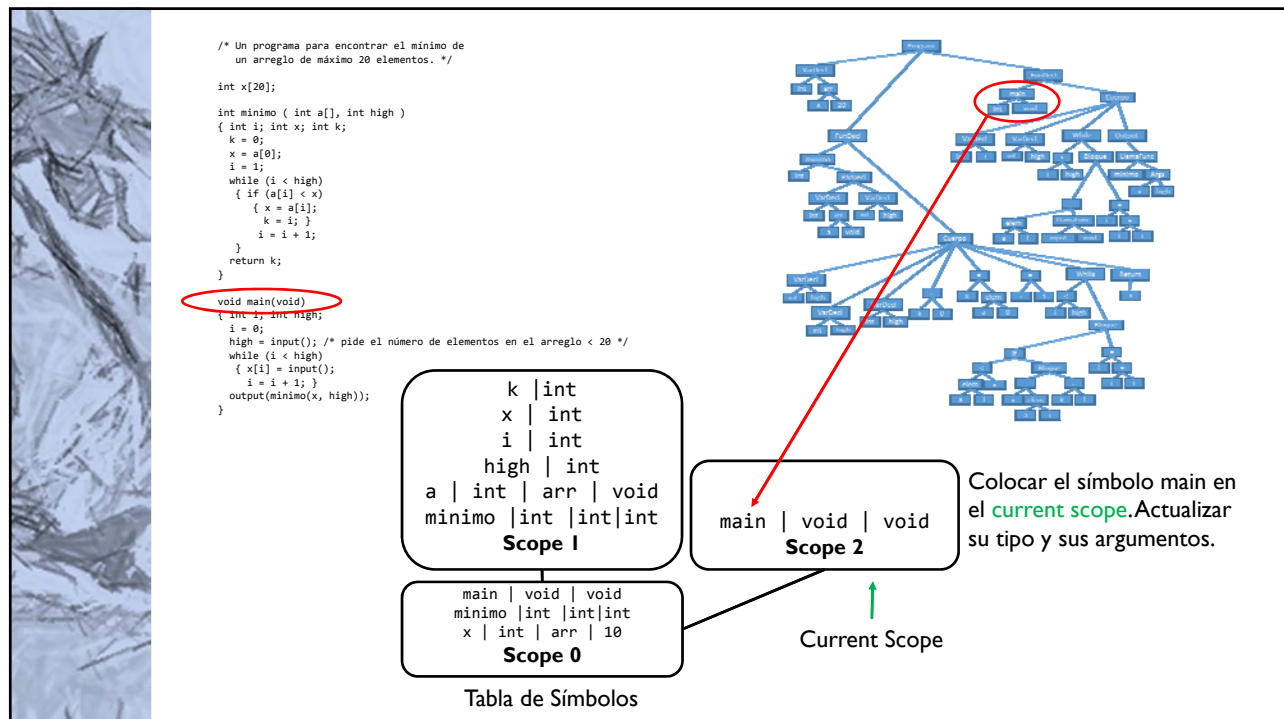
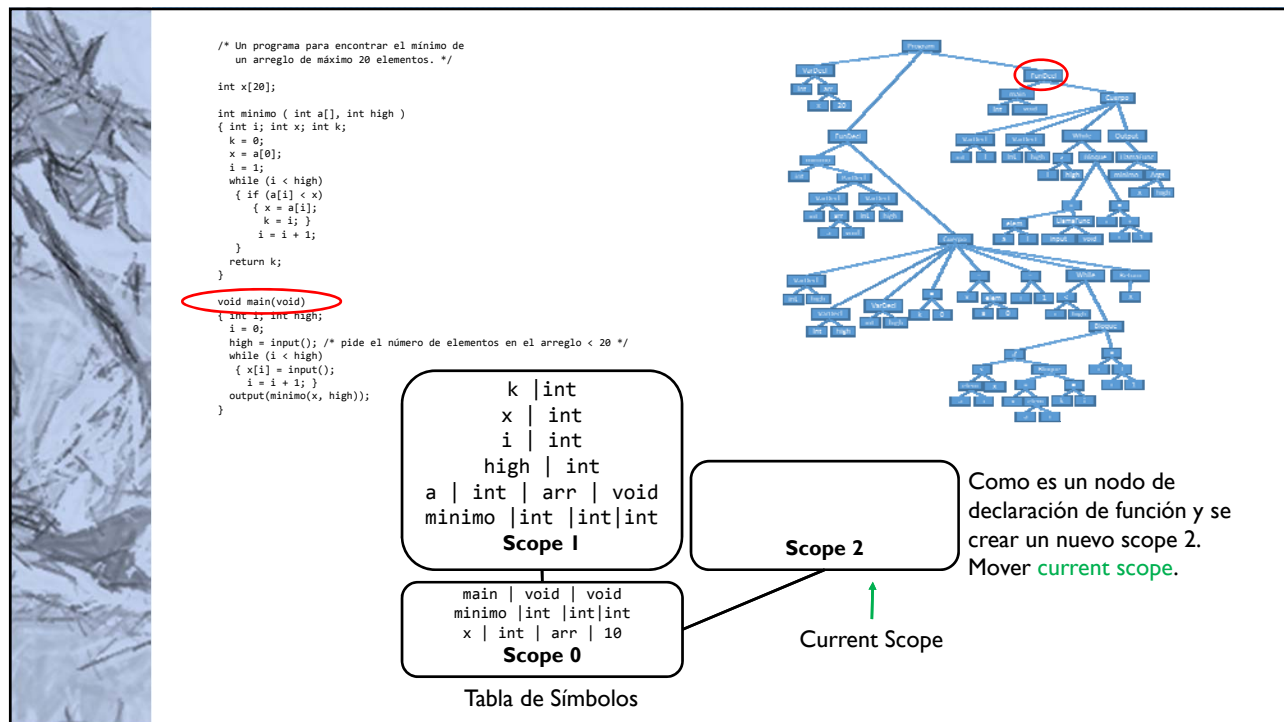
k   int
x   int
i   int
high   int
a   int   arr   void
minimo   int   int   int
<b>Scope I</b>

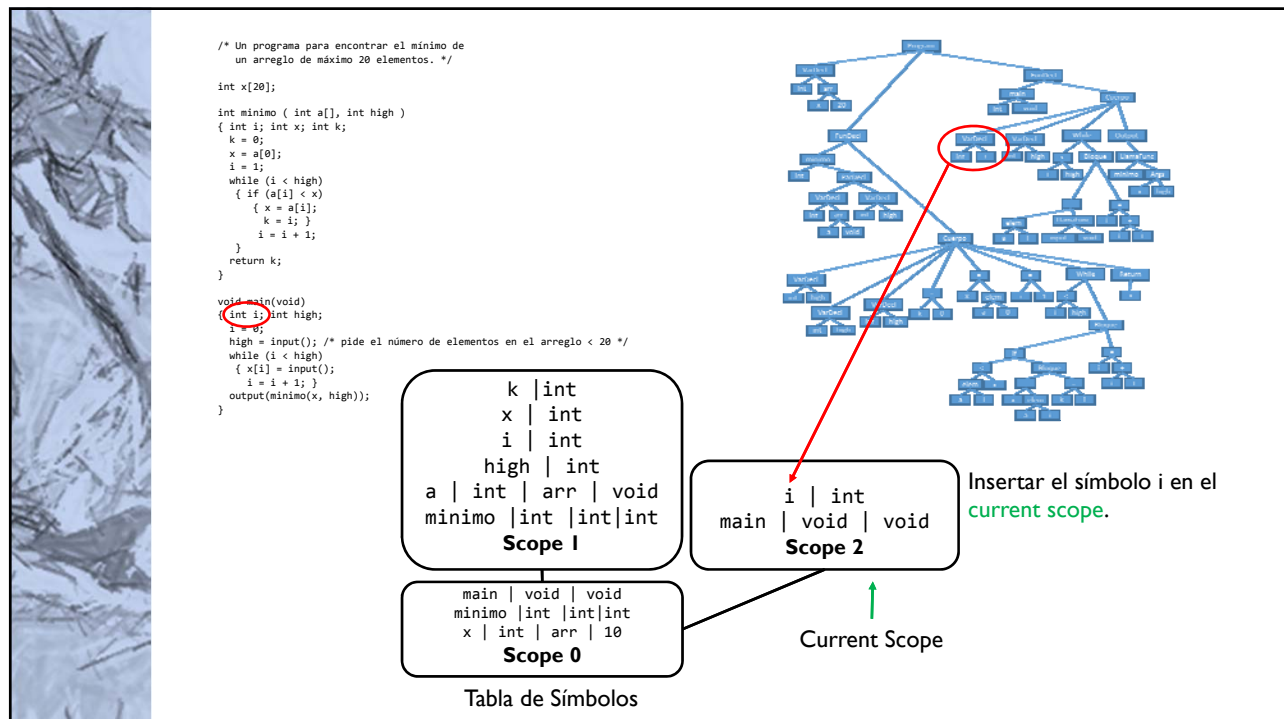
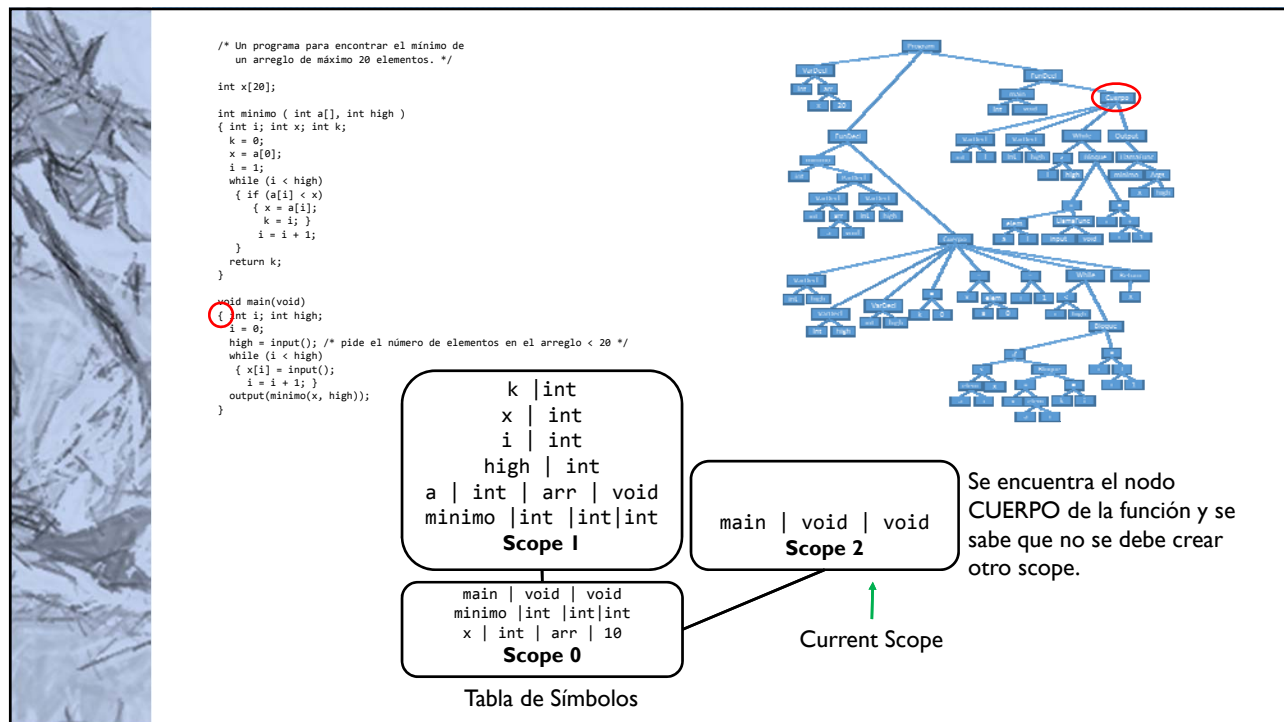
  

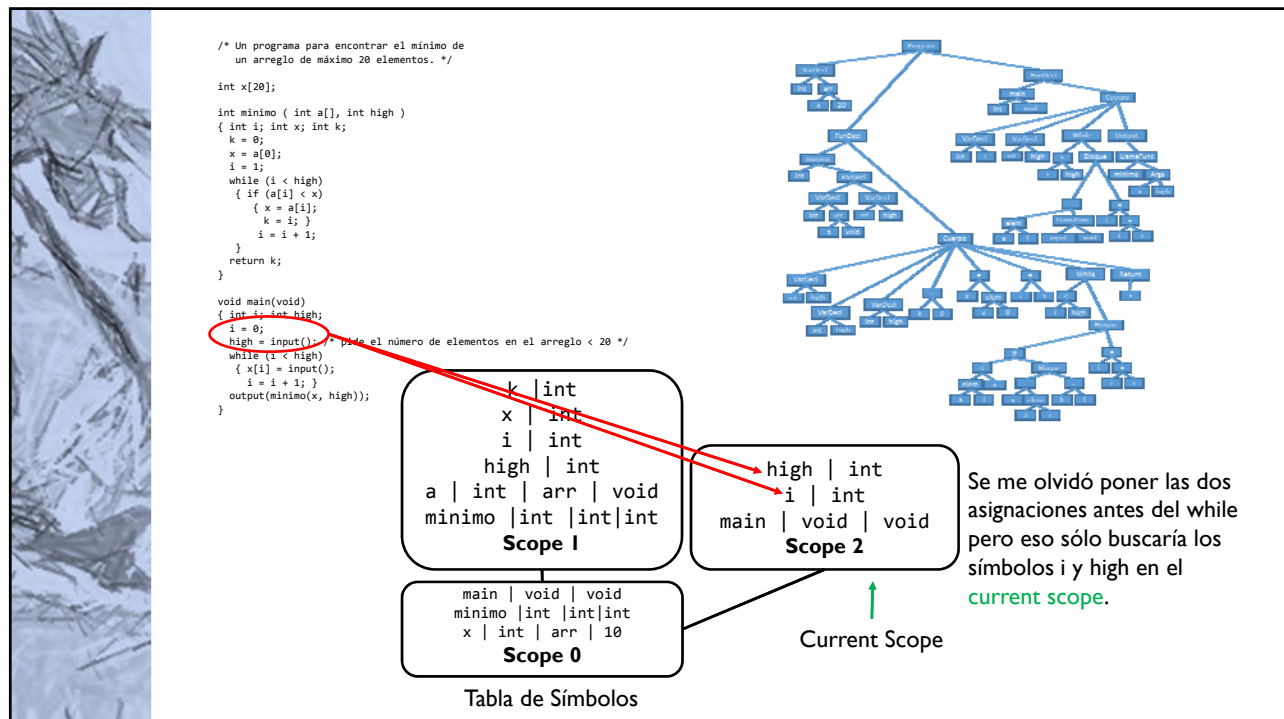
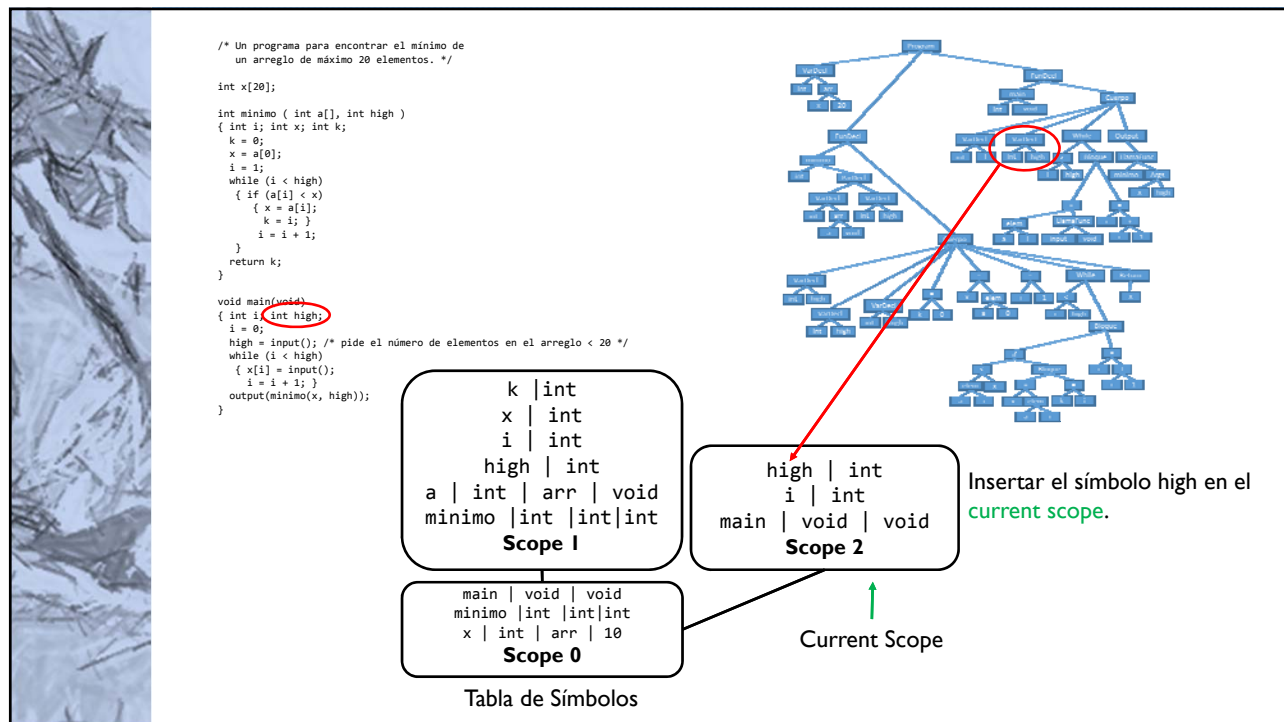
main   void   void
minimo   int   int   int
x   int   arr   10
<b>Scope 0</b>

← Current Scope

Tabla de Símbolos







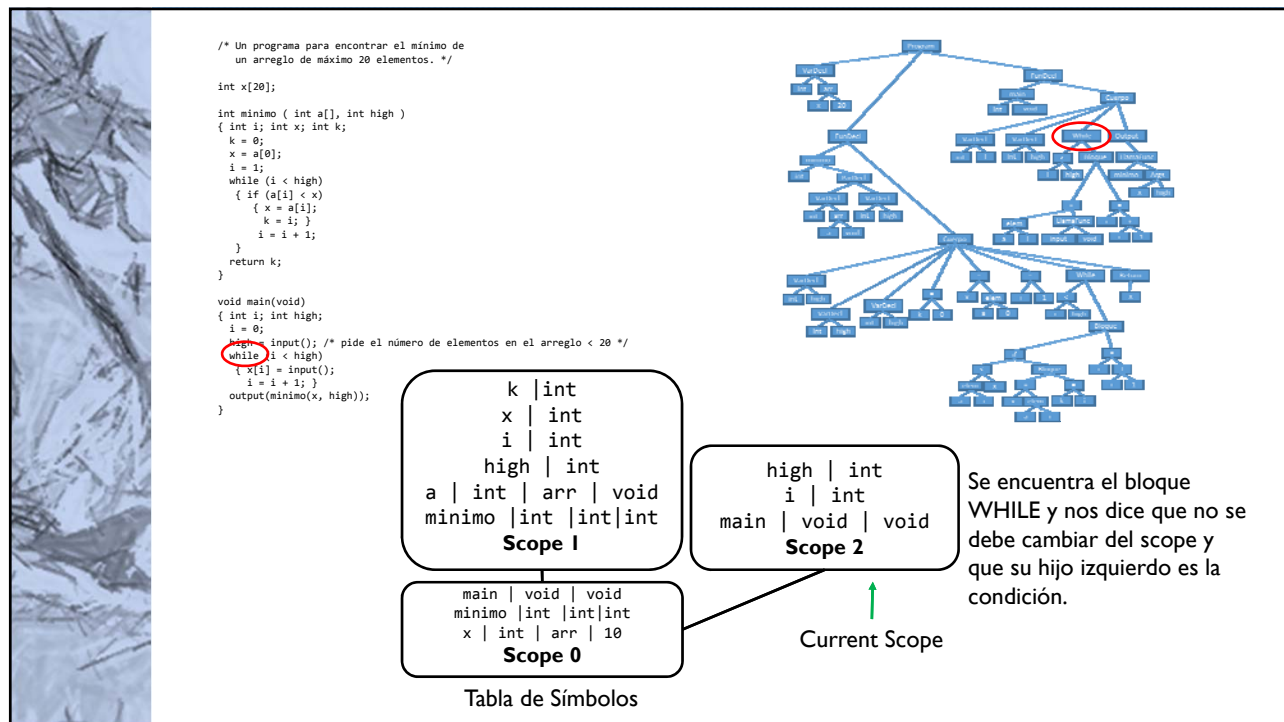


Tabla de Símbolos

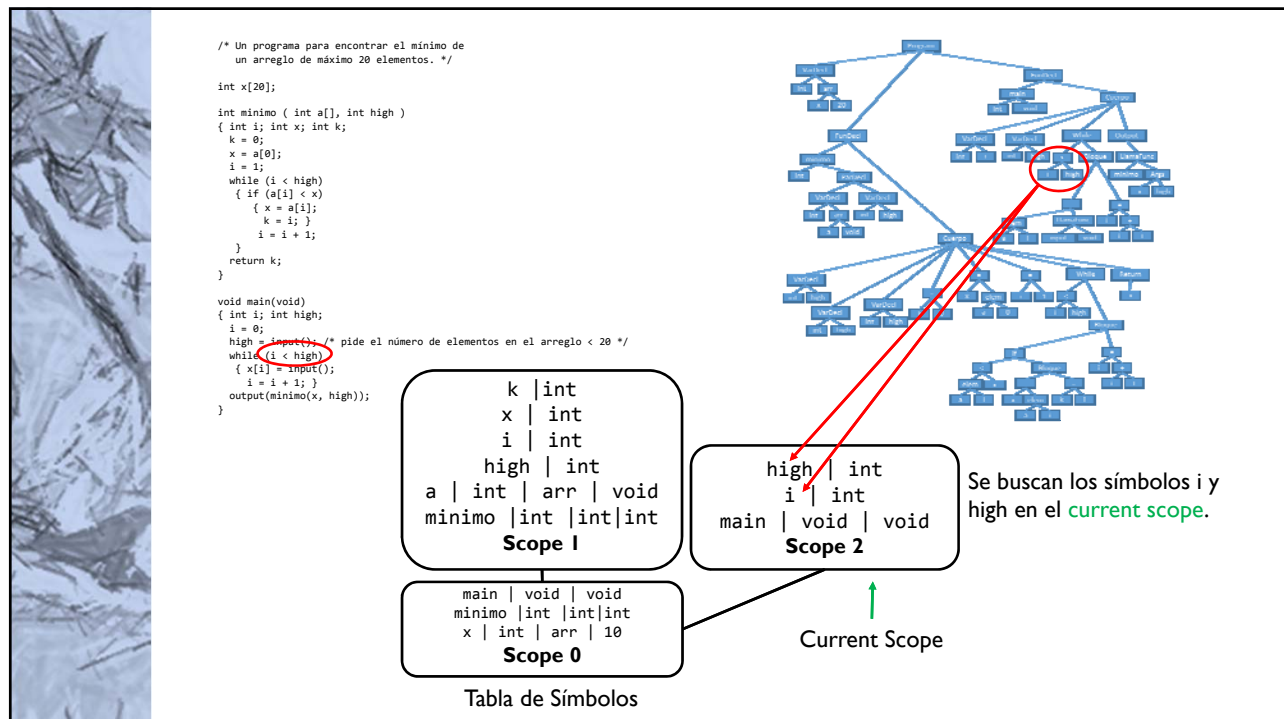
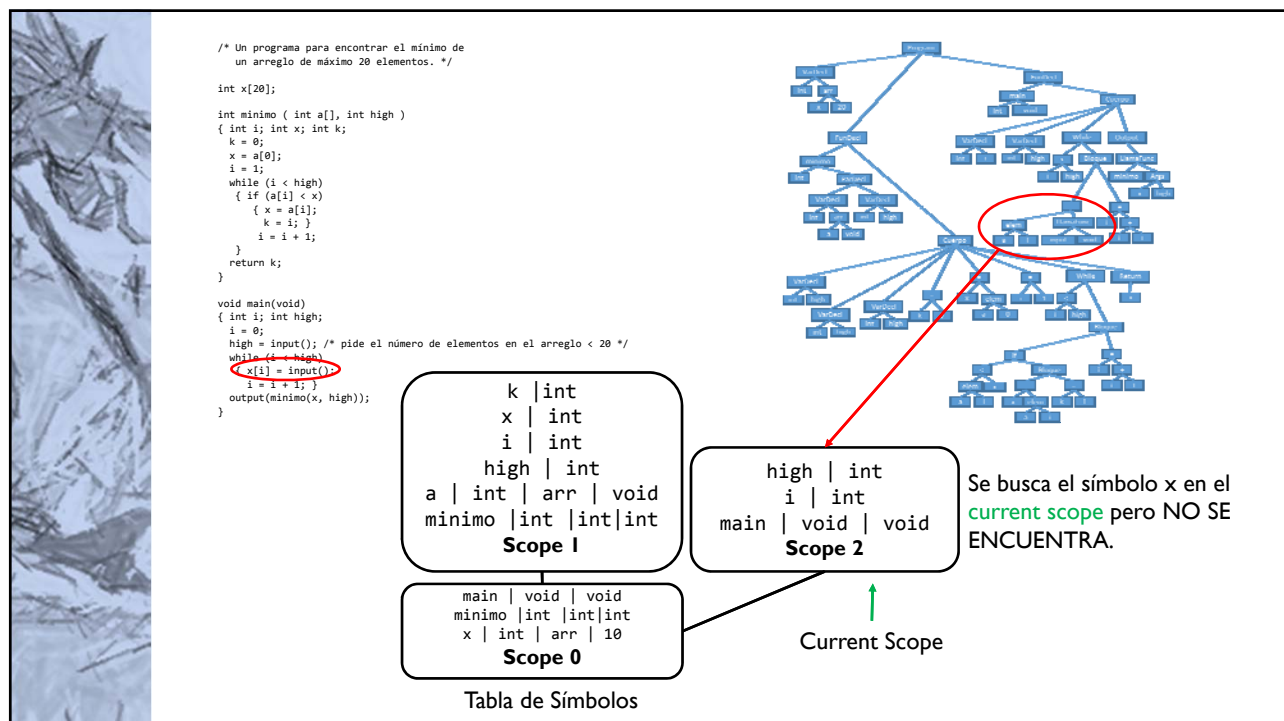
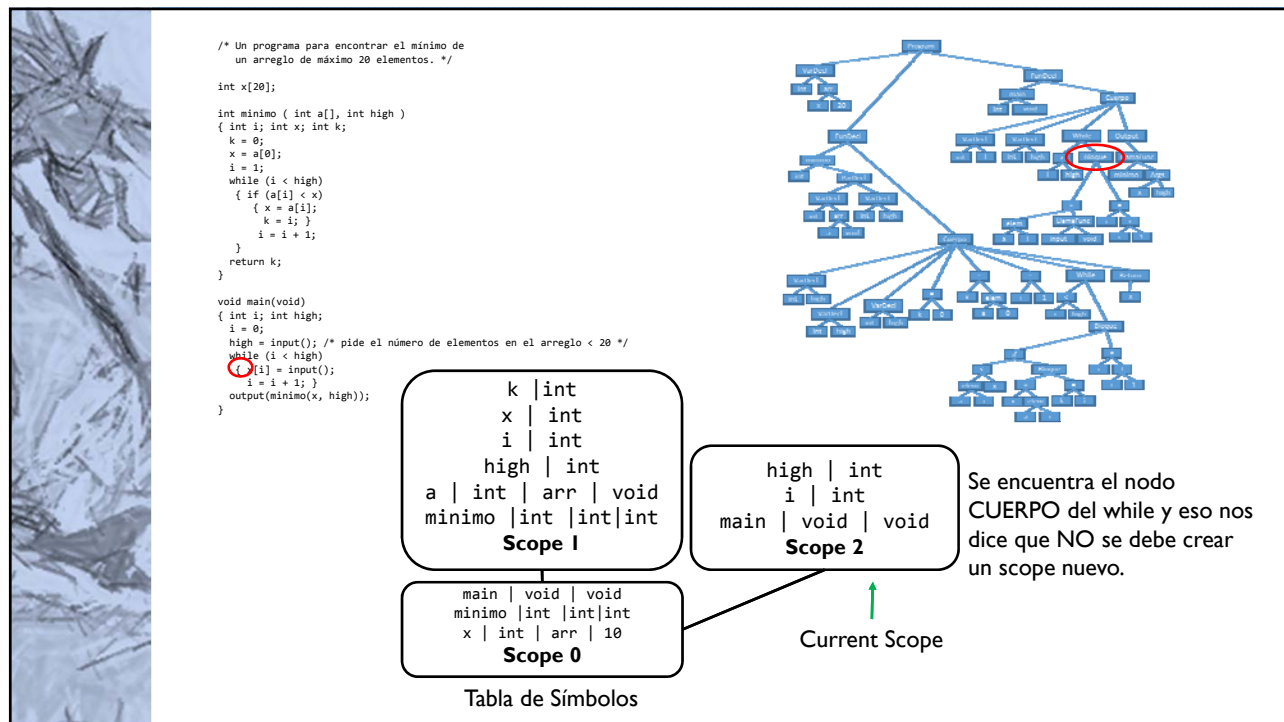
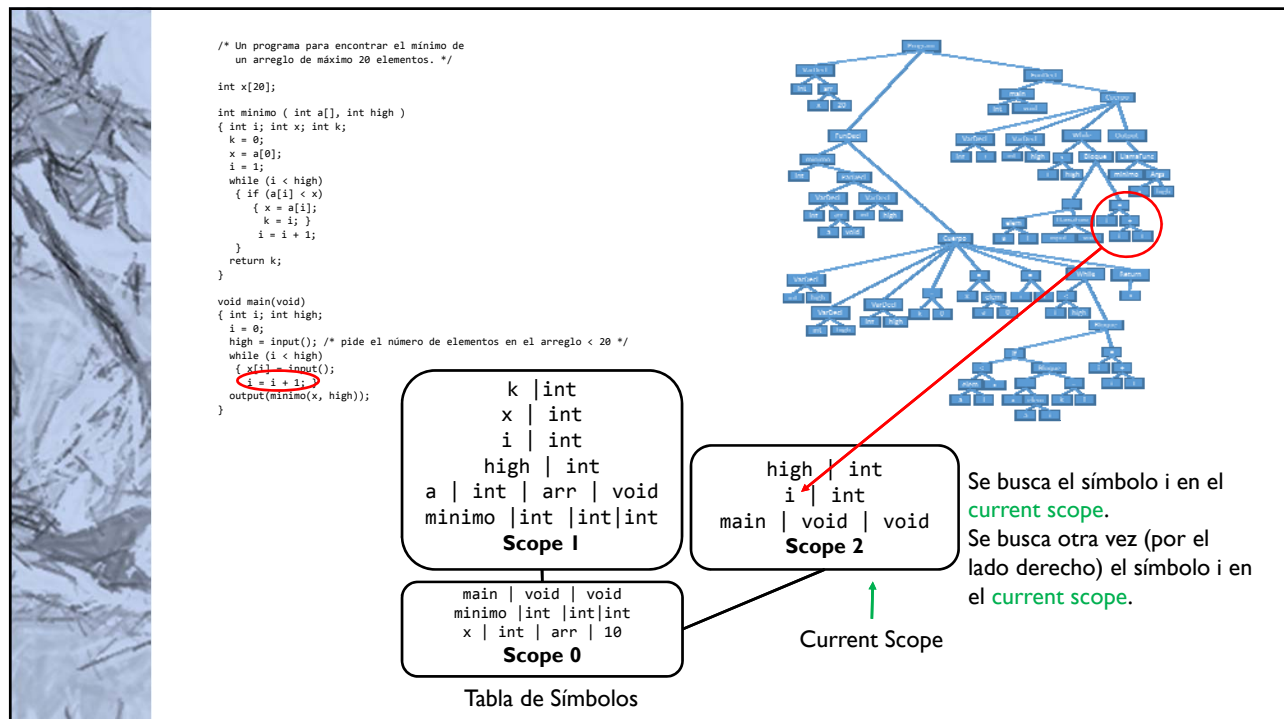
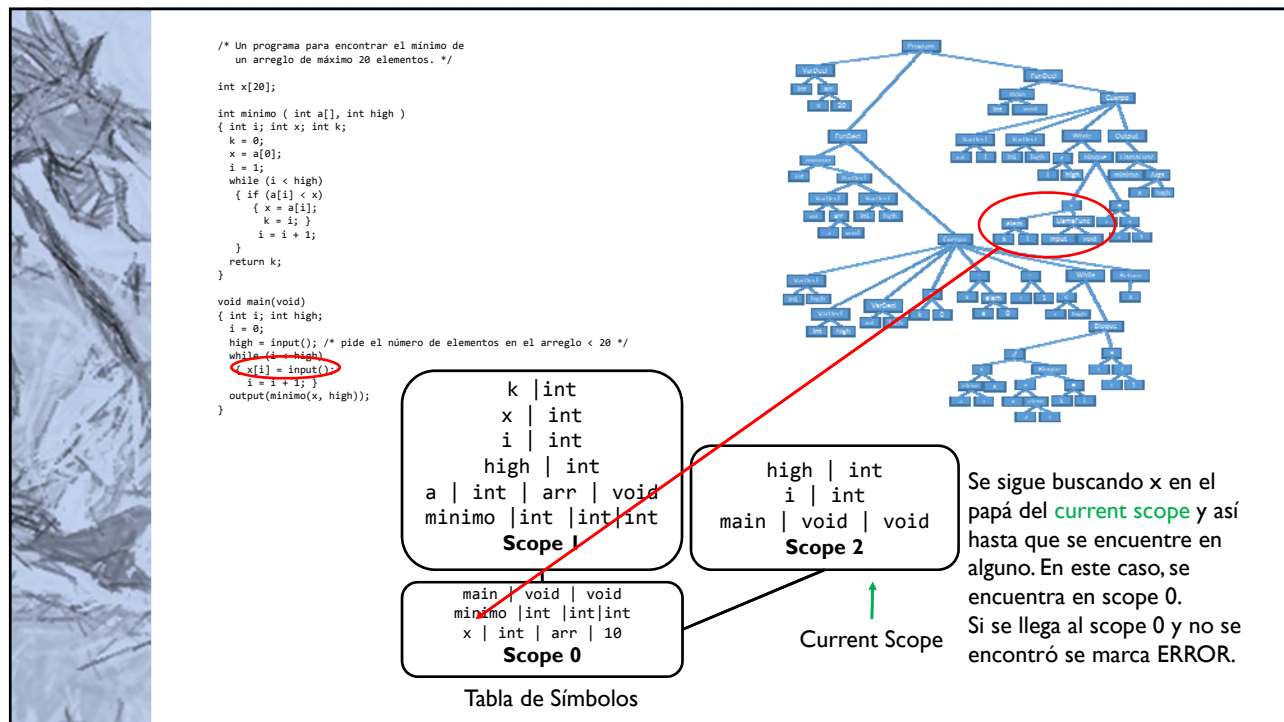
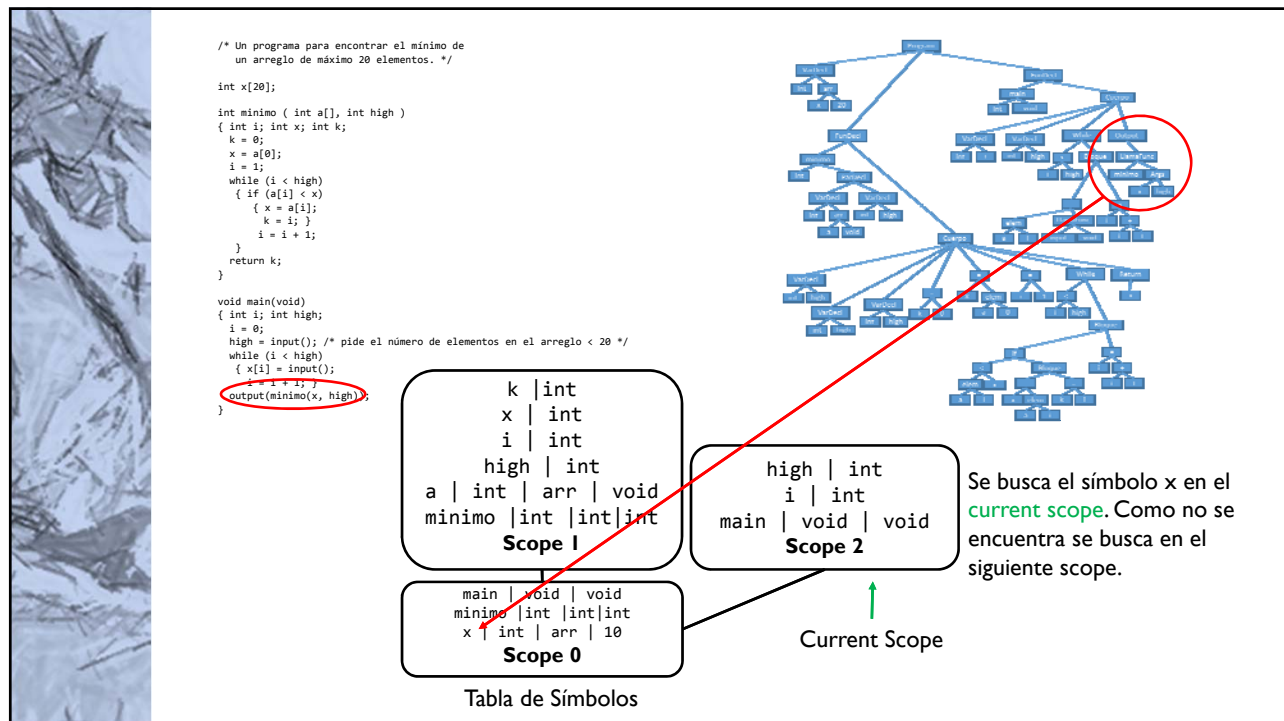
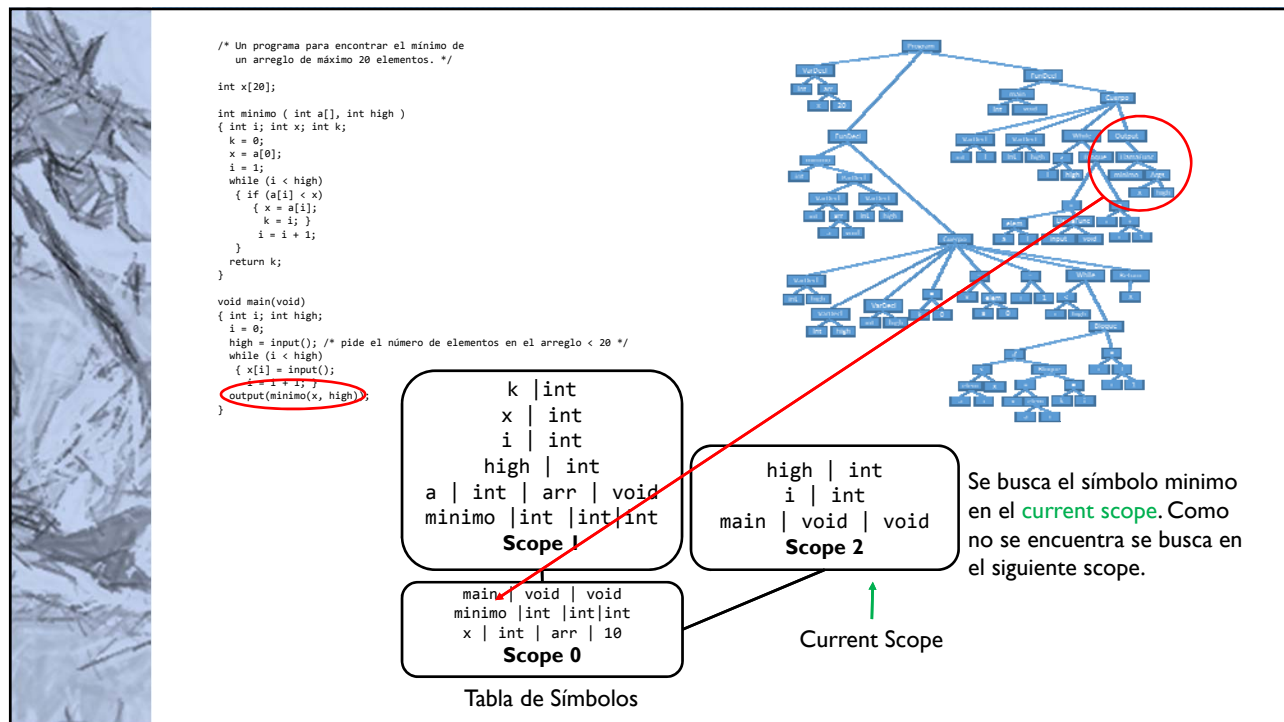


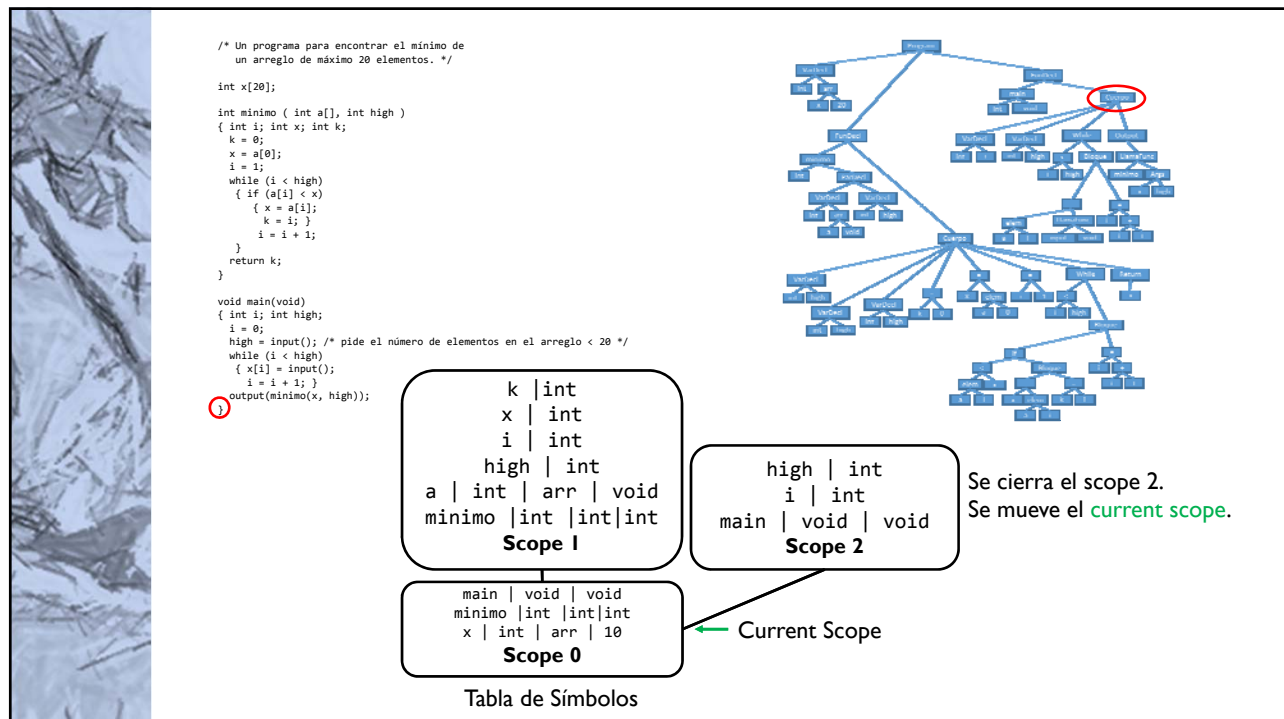
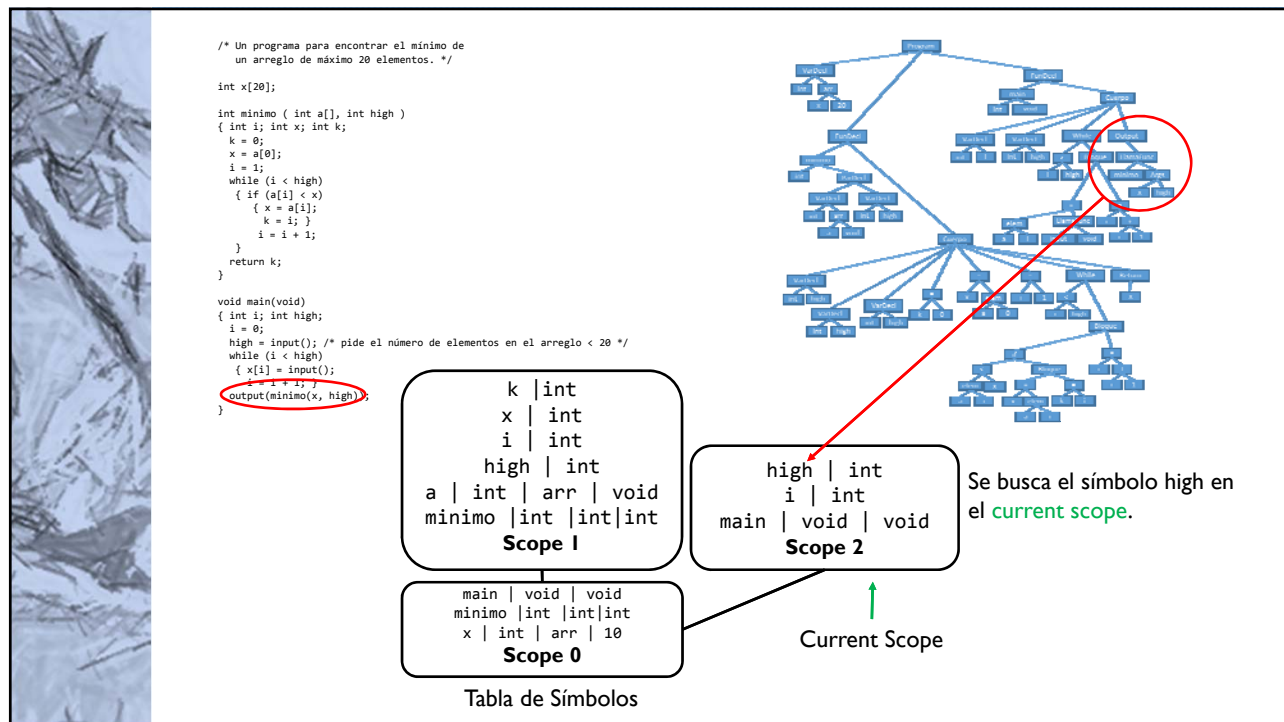
Tabla de Símbolos











```

/* Un programa para encontrar el mínimo de
un arreglo de máximo 20 elementos. */

int x[20];

int minimo ( int a[], int high )
{ int i; int x; int k;
  k = 0;
  x = a[0];
  i = 1;
  while ( i < high )
  { if ( a[i] < x )
    { x = a[i];
      k = i;
    }
    i = i + 1;
  }
  return k;
}

void main(void)
{ int i; int high;
  i = 0;
  high = input(); /* pide el número de elementos en el arreglo < 20 */
  while ( i < high )
  { x[i] = input();
    i = i + 1;
  }
  output(minimo(x, high));
}

```

k		int
x		int
i		int
high		int
a		int   arr   void
minimo		int   int   int
<b>Scope 1</b>		

high		int
i		int
main		void   void
<b>Scope 2</b>		

Se cierra el scope 0.

main		void   void
minimo		int   int   int
x		int   arr   10
<b>Scope 0</b>		

Tabla de Símbolos

## Referencias

- Alan G. Labouseur.  
<http://labouseur.com/courses/compilers/AST-and-Symbol-Table.pdf>. (consultado el 24-oct-2018)