Contract Code

```
using Neo.SmartContract.Framework;
using Neo.SmartContract.Framework.Services.Neo;

public class Sum : SmartContract
{
    public static int Main(int a, int b)
    {
        return a + b;
    }
}
```

MSIL

main function IL code

```
IL_0000 Nop
IL_0001 Ldarg_0
IL_0002 Ldarg_1
IL_0003 Add
IL_0004 Stloc_0
IL_0005 Br_S
IL_0007 Ldloc_0
IL_0008 Ret
```

neo.compiler

```
hex:53-C5-6B-6C-76-6B-00-52-7A-C4-6C-76-6B-51-52-7A-C4-61-6C-76-6B-00-
C3-6C-76-6B-51-C3-93-6C-76-6B-52-52-7A-C4-62-03-00-6C-76-6B-52-C3-61-6C-75-66
```

```
asm:
PUSH4
PUSH3
RET
PUSH3
NEWARRAY
TOTALSTACK
FROMALSTACK
DUP
TOALTSTACK
PUSH0
PUSH2
ROLL
SETITEM
FROMALSTACK
DUP
TOTALSTACK
PUSH1
PUSH2
ROLL
SETITEM
NOP
FROMALSTACK
DUP
TOTALSTACK
PUSH0
```

```
PICKITEM
FROMALSTACK
DUP
TOTALSTACK
PUSH1
PICKITEM
ADD
FROMALSTACK
DUP
TOTALSTACK
PUSH2
PUSH2
ROLL
SETITEM
JMP
FROMALSTACK
DUP
TOTALSTACK
PUSH2
PICKITEM
NOP
FROMALSTACK
DROP
ret
```

neo.vm

OpCode	Evaluation Stack	AltStack	InvoStack	Comment	phase
LoadScript			script:ExecContext	engine load script	push parameters

LoadScript			[script:ExecContext, param:ExecContext]	engine load parameters
				sb.EmitPush(4); // 对应形参 b
				sb.EmitPush(3); // 对应形参 a
				engine.LoadScript(sb.ToArray());	

PUSH4	4:integer		same		
PUSH3	[4:integer, 3:integer]		same		
RET	[4:integer, 3:integer]		[script:ExecContext]	OpCode opcode = CurrentContext.InstructionPointer >= CurrentContext.Script.Length ? OpCode.RET : (OpCode)CurrentContext.OpReader.ReadByte();	
PUSH3	[4:integer, 3:integer, 3:integer]		same	why push 3? because has two paramters. You can find details in the _insertBeginCode of neo.compiler	please see _insertBeginCode of neo-compiler/neon/MSIL/Conv_Common.cs

OpCode	Evaluation Stack	AltStack	InvoStack	Comment	phase
NEWARRAY	[4:integer, 3:integer, [false,false,false]:Array]		same	1. POP -> size 2. new List<StackItem>(count); 3. EvaluationStack.Push(new Types.Array(items));	
TOTALSTACK	[4:integer, 3:integer]	[false,false,false]:Array	same	移除计算栈栈顶的元素，并将其压入备用栈。	
FROMALSTACK	[4:integer, 3:integer, [false,false,false]:Array]		same	移除备用栈栈顶的元素，并将其压入计算栈。 //set param:0	
DUP	[4:integer, 3:integer, [false,false,false]:Array, [false,false,false]:Array]		same		
TOALTSTACK	[4:integer, 3:integer, [false,false,false]:Array]	[false,false,false]:Array	same		
PUSH0	[4:integer, 3:integer, [false,false,false]:Array,0:ByteArray]	[false,false,false]:Array	same		
PUSH2	[4:integer, 3:integer, [false,false,false]:Array,0:ByteArray, 2:integer]	[false,false,false]:Array	same		
ROLL	[4:integer, [false,false,false]:Array,0:ByteArray, 3:integer]	[false,false,false]:Array	same	移除计算栈栈顶的元素n，并将剩余的索引为n的元素移动到栈顶。	
SETITEM	[4:integer]	[3,false,false]:Array	same	items[index] = newItem; 1. POP -> newItem 2. POP -> index 3. POP -> items	
FROMALSTACK	[4:integer, [3:integer,false,false]:Array]		same	// set param 1	

OpCode	Evaluation Stack	AltStack	InvoStack	Comment	phase
DUP	[4:integer, [3:integer,false,false]:Array, [3:integer,false,false]:Array]		same		
TOTALSTACK	[4:integer, [3:integer,false,false]:Array]	[3:integer,false,false]:Array	same		
PUSH1	[4:integer, [3:integer,false,false]:Array, 1:integer]	[3:integer,false,false]:Array	same		
PUSH2	[4:integer, [3:integer,false,false]:Array, 1:integer, 2:integer]	[3:integer,false,false]:Array	same		
ROLL	[[3:integer,false,false]:Array, 1:integer, 4:integer]	[3:integer,false,false]:Array	same	输入： Xn Xn-1 ... X2 X1 X0 n 输出： Xn-1 ... X2 X1 X0 Xn	
SETITEM		[3:integer, 4:integer,false]: Array	same	items[index] = newItem; 1. POP -> newItem 2. POP -> index 3. POP -> items	
NOP			same		begin translate MSIL to neo.vm opcode
FROMALSTACK	[3:integer, 4:integer,false]:Array		same		ldarg_0
DUP	[[3:integer, 4:integer,false]:Array,[3:integer, 4:integer,false]:Array]		same		
TOTALSTACK	[3:integer, 4:integer,false]:Array	[3:integer, 4:integer,false]: Array	same		
PUSH0	[[3:integer, 4:integer,false]:Array,0:ByteArray]	[3:integer, 4:integer,false]: Array	same		

OpCode	Evaluation Stack	AltStack	InvoStack	Comment	phase
PICKITEM	[3:integer]	[3:integer, 4:integer,false]: Array	same	EvaluationStack.Push(items[index]); 1. POP -> index 2. POP -> item 3. items -> item.GetArray() 4. EvaluationStack.Push(items[index]);	
FROMALSTACK	[3:integer, [3:integer, 4:integer,false]:Array]		same		ldarg_1
DUP	[3:integer, [3:integer, 4:integer,false]:Array, , [3:integer, 4:integer,false]:Array]		same		
TOTALSTACK	[3:integer, [3:integer, 4:integer,false]:Array]	[3:integer, 4:integer,false]: Array	same		
PUSH1	[3:integer, [3:integer, 4:integer,false]:Array,1:integer]	[3:integer, 4:integer,false]: Array	same		
PICKITEM	[3:integer, 4:integer]	[3:integer, 4:integer,false]: Array	same		
ADD	[7:integer]	[3:integer, 4:integer,false]: Array	same		IL:ADD
FROMALSTACK	[7:integer, [3:integer, 4:integer,false]:Array]		same		stloc_0
DUP	[7:integer, [3:integer, 4:integer,false]:Array, [3:integer, 4:integer,false]:Array]		same		

OpCode	Evaluation Stack	AltStack	InvoStack	Comment	phase
TOTALSTACK	[7:integer, [3:integer, 4:integer,false]:Array]	[3:integer, 4:integer,false]:Array	same		
PUSH2	[7:integer, [3:integer, 4:integer,false]:Array,2:integer]	[3:integer, 4:integer,false]:Array	same		
PUSH2	[7:integer, [3:integer, 4:integer,false]:Array,2:integer, 2:integer]	[3:integer, 4:integer,false]:Array	same		
ROLL	[[3:integer, 4:integer,false]:Array,2:integer, 7:integer]	[3:integer, 4:integer,false]:Array	same	input： Xn Xn-1 ... X2 X1 X0 output： Xn-1 ... X2 X1 X0 Xn	
SETITEM		[3:integer, 4:integer, 7:integer]:Array	same	items[index] = newItem; 1. POP -> newItem 2. POP -> index 3. POP -> items	
JMP		[3:integer, 4:integer, 7:integer]:Array	same	why read 003:int16? uesless	Br_S
FROMALSTACK	[3:integer, 4:integer, 7:integer]:Array		same		Ldloc_0
DUP	[[3:integer, 4:integer, 7:integer]:Array, [3:integer, 4:integer, 7:integer]:Array]		same		
TOTALSTACK	[3:integer, 4:integer, 7:integer]:Array	[3:integer, 4:integer, 7:integer]:Array	same		
PUSH2	[[3:integer, 4:integer, 7:integer]:Array, 2:integer]		same		

OpCode	Evaluation Stack	AltStack	InvoStack	Comment	phase
PICKITEM	[7:integer]	[3:integer, 4:integer,false]: Array	same	EvaluationStack.Push(items[index]); 1. POP -> index 2. POP -> item 3. items -> item.GetArray() 4. EvaluationStack.Push(items[index]);	
NOP	[7:integer]	[3:integer, 4:integer,false]: Array	same		end translate MSIL to neo.vm opcode
FROMALSTACK	[7:integer, [3:integer, 4:integer,false]:Array]		same	//endcode	_insertEndCode
DROP	[7:integer]		same	移除计算栈栈顶的元素。	_insertEndCode
ret				移除调用栈的顶部元素，并使程序在调用栈的下一帧中继续执行。如果调用栈 为空，则虚拟机进入停机状态。	ret