Image from http://www.szzljy.com

Image from http://www.szzljy.com

Image from http://www.szzljy.com

Image from http://www.szzljy.com

http://www.szzljy.comBy Alejandro Santander in collaboration with Leo Arias.

By Alejandro Santander in collaboration with Leo Arias.

Alejandro SantanderLeo AriasYou're on the road, driving fast in your rare, fully restored 1969 Mustang Mach 1. The sunlight shimmers on the all-original, gorgeous plated rims. It's just you, the road, the desert, and the never-ending chase of the horizon. Perfection!

In the blink of an eye, your 335 hp beast is engulfed in white smoke, as if transformed into a steam locomotive, and you're forced to stop on the side of the road. With determination, you pop the hood, only to realize that you have absolutely no idea what you're looking at. How does this damn machine work? You grab your phone and discover that you have no signal.

Image from http://www.mustangandfords.com

Image from http://www.mustangandfords.com

Image from http://www.mustangandfords.com

http://www.mustangandfords.comCould this perhaps be an analogy for your current knowledge of dApp development? In the analogy, the Mustang is your set of smart contracts, the rims are all those well-thought-out little details and the ❤ you put into them. And the popping of the hood is you looking into your contract's EVM bytecode and having absolutely no idea what's going on.

If this sounds familiar, then not to worry! The purpose of this series of articles is to deconstruct a simple Solidity contract, look at its bytecode, and break it apart into identifiable structures down to the lowest level. We'll pop the hood on Solidity. By the end of the series, you should feel comfortable when looking at or debugging EVM bytecode. The whole point of the series is to demystify the EVM bytecode produced by the Solidity compiler. And it's really much simpler than it seems.

Note: This series is aimed at developers who already feel comfortable with and have experience in developing Solidity contracts, but want to understand how things work at a slightly deeper/lower level—that is, how Solidity is translated into EVM bytecode by the Solidity compiler, and how the EVM executes such bytecode. If you aren't there just yet, I recommend reading this great introduction by Facu Spagnuolo: A Gentle Introduction to Ethereum Programming.

Note: This series is aimed at developers who already feel comfortable with and have experience in developing Solidity contracts, but want to understand how things work at a slightly deeper/lower level—that is, how Solidity is translated into EVM bytecode by the Solidity compiler, and how the EVM executes such bytecode. If you aren't there just yet, I recommend reading this great introduction by Facu Spagnuolo: A Gentle Introduction to Ethereum Programming.

A Gentle Introduction to Ethereum ProgrammingHere's the contract we'll deconstruct:

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

```
pragma solidity
^
0.4.24
;
contract
BasicToken
{
uint256
totalSupply_;
mapping
(address
=>
uint256
) balances;
constructor
(uint256
_initialSupply
) public
{
totalSupply_ =
_initialSupply;
balances[msg
.sender
] =
_initialSupply;
}
functiontotalSupply
() public
view
returns
(uint256
) {
return
totalSupply_;
}
```

functiontransfer

(address

_to

, uint256

_value

) public

returns

(bool

) {

require

(_to !=

address

(0

));

require

(_value <=

balances[msg

.sender

]);

balances[msg

.sender

] =

balances[msg

.sender

] –

_value;

balances[_to] =

balances[_to] +

_value;

return

true

;

}

functionbalanceOf

(address

_owner

) public

view

returns

(uint256

) {

return

balances[_owner];

}

}

view raw

[

BasicToken.sol

](https://gist.github.com/ajsantander/dce951a95e7608bc29d7f5deeb6e2ecf#file-basictoken-sol)

hosted with ❤ by GitHub

pragma solidity

^

0.4.24

;

contract

BasicToken

```
{
uint256
totalSupply_;
mapping
(address
=>
uint256
) balances;
constructor
(uint256
_initialSupply
) public
{
totalSupply_ =
_initialSupply;
balances[msg
.sender
] =
_initialSupply;
}
functiontotalSupply
() public
view
returns
(uint256
) {
return
totalSupply_;
}
functiontransfer
(address
_to
, uint256
_value
) public
returns
(bool
) {
require
(_to !=
address
(0
));
require
(_value <=
balances[msg
.sender
]);
balances[msg
.sender
] =
balances[msg
.sender
] –
_value;
balances[_to] =
balances[_to] +
_value;
return
true
```

;

}

functionbalanceOf

(address

_owner

) public

view

returns

(uint256

) {

return

balances[_owner];

}

}

view raw

[

BasicToken.sol

](https://gist.github.com/ajsantander/dce951a95e7608bc29d7f5deeb6e2ecf#file-basictoken-sol)

hosted with ❤ by GitHub

pragma solidity

^

0.4.24

;

contract

BasicToken

{

uint256

totalSupply_;

mapping

(address

=>

uint256

) balances;

constructor

(uint256

_initialSupply

) public

{

totalSupply_ =

_initialSupply;

balances[msg

.sender

] =

_initialSupply;

}

functiontotalSupply

() public

view

returns

(uint256

) {

return

totalSupply_;

}

functiontransfer

(address

_to
, uint256
_value
) public
returns
(bool
) {
require
(_to !=
address
(0
));
require
(_value <=
balances[msg
.sender
]);
balances[msg
.sender
] =
balances[msg
.sender
] –
_value;
balances[_to] =
balances[_to] +
_value;
return
true
;
}
functionbalanceOf
(address
_owner
) public
view
returns
(uint256
) {
return
balances[_owner];
}
}
This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.
Learn more about bidirectional Unicode characters
[Show hidden characters
]()
pragma solidity
^
0.4.24
;
contract
BasicToken
{
uint256
totalSupply_;
mapping
(address
=>
uint256

```
) balances;
constructor
(uint256
_initialSupply
) public
{
totalSupply_ =
_initialSupply;
balances[msg
.sender
] =
_initialSupply;
}
functiontotalSupply
() public
view
returns
(uint256
) {
return
totalSupply_;
}
functiontransfer
(address
_to
, uint256
_value
) public
returns
(bool
) {
require
(_to !=
address
(0
));
require
(_value <=
balances[msg
.sender
]);
balances[msg
.sender
] =
balances[msg
.sender
] –
_value;
balances[_to] =
balances[_to] +
_value;
return
true
;
}
functionbalanceOf
(address
_owner
) public
view
```

returns

(uint256

) {

return

balances[_owner];

}

}

pragma solidity

^

0.4.24

;

contract

BasicToken

{

uint256

totalSupply_;

mapping

(address

=>

uint256

) balances;

constructor

(uint256

_initialSupply

) public

{

totalSupply_ =

_initialSupply;

balances[msg

.sender

] =

_initialSupply;

}

functiontotalSupply

() public

view

returns

(uint256

) {

return

totalSupply_;

}

functiontransfer

(address

_to

, uint256

_value

) public

returns

(bool

) {

require

(_to !=

address

(0

));

```
require
(_value <=
balances[msg
.sender
]);
balances[msg
.sender
] =
balances[msg
.sender
] –
_value;
balances[_to] =
balances[_to] +
_value;
return
true
;
}
functionbalanceOf
(address
_owner
) public
view
returns
(uint256
) {
return
balances[_owner];
}
}
```



```
pragma solidity
^
0.4.24
;
contract
BasicToken
{
uint256
totalSupply_;
mapping
(address
=>
uint256
) balances;
constructor
(uint256
_initialSupply
) public
{
totalSupply_ =
_initialSupply;
balances[msg
.sender
] =
_initialSupply;
```

}
functiontotalSupply
() public
view
returns
(uint256
) {
return
totalSupply_;
}
functiontransfer
(address
_to
, uint256
_value
) public
returns
(bool
) {
require
(_to !=
address
(0
));
require
(_value <=
balances[msg
.sender
]);
balances[msg
.sender
] =
balances[msg
.sender
] –
_value;
balances[_to] =
balances[_to] +
_value;
return
true
;
}
functionbalanceOf
(address
_owner
) public
view
returns
(uint256
) {
return
balances[_owner];
}
}
This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.
Learn more about bidirectional Unicode characters
[Show hidden characters
]()
pragma solidity

^

0.4.24

;

contract

BasicToken

{

uint256

totalSupply_;

mapping

(address

=>

uint256

) balances;

constructor

(uint256

_initialSupply

) public

{

totalSupply_ =

_initialSupply;

balances[msg

.sender

] =

_initialSupply;

}

functiontotalSupply

() public

view

returns

(uint256

) {

return

totalSupply_;

}

functiontransfer

(address

_to

, uint256

_value

) public

returns

(bool

) {

require

(_to !=

address

(0

));

require

(_value <=

balances[msg

.sender

]);

balances[msg

.sender

] =

balances[msg

.sender

] –

_value;

balances[_to] =

balances[_to] +

_value;

return

true

;

}

functionbalanceOf

(address

_owner

) public

view

returns

(uint256

) {

return

balances[_owner];

}

}

Learn more about bidirectional Unicode characters

pragma solidity

^

0.4.24

;

contract

BasicToken

{

uint256

totalSupply_;

mapping

(address

=>

uint256

) balances;

constructor

(uint256

_initialSupply

) public

{

totalSupply_ =

_initialSupply;

balances[msg

.sender

] =

_initialSupply;

}

functiontotalSupply

() public

view

returns

```
(uint256
) {
return
totalSupply_;
}
functiontransfer
(address
_to
, uint256
_value
) public
returns
(bool
) {
require
(_to !=
address
(0
));
require
(_value <=
balances[msg
.sender
]);
balances[msg
.sender
] =
balances[msg
.sender
] –
_value;
balances[_to] =
balances[_to] +
_value;
return
true
;
}
functionbalanceOf
(address
_owner
) public
view
returns
(uint256
) {
return
balances[_owner];
}
}
pragma solidity
^
0.4.24
;
contract
BasicToken
{
uint256
totalSupply_;
mapping
```

```
(address
=>
uint256
) balances;
constructor
(uint256
_initialSupply
) public
{
totalSupply_ =
_initialSupply;
balances[msg
.sender
] =
_initialSupply;
}
functiontotalSupply
() public
view
returns
(uint256
) {
return
totalSupply_;
}
functiontransfer
(address
_to
, uint256
_value
) public
returns
(bool
) {
require
(_to !=
address
(0
));
require
(_value <=
balances[msg
.sender
]);
balances[msg
.sender
] =
balances[msg
.sender
] –
_value;
balances[_to] =
balances[_to] +
_value;
return
true
;
}
functionbalanceOf
(address
```

_owner
) public
view
returns
(uint256
) {
return
balances[_owner];
}
}
pragma solidity
^
0.4.24
;
pragma solidity
^
0.4.24
;
pragma solidity
^
0.4.24
contract
BasicToken
{
contract
BasicToken
{
contract
BasicToken
uint256
totalSupply_;
uint256
totalSupply_;
uint256
mapping
(address
=>
uint256
) balances;
mapping
(address
=>
uint256
) balances;
mapping
address
=>
uint256
constructor
(uint256
_initialSupply
) public
{
constructor
(uint256
_initialSupply
) public
{
constructor
uint256

_initialSupply
public
totalSupply_ =
_initialSupply;
totalSupply_ =
_initialSupply;
=
balances[msg
.sender
] =
_initialSupply;
balances[msg
.sender
] =
_initialSupply;
msg
sender
=
}
}
functiontotalSupply
() public
view
returns
(uint256
) {
functiontotalSupply
() public
view
returns
(uint256
) {
functiontotalSupply
totalSupply
public
view
returns
uint256
return
totalSupply_;
return
totalSupply_;
return
}
}
functiontransfer
(address
_to
, uint256
_value
) public
returns
(bool
) {
functiontransfer
(address
_to
, uint256
_value
) public

returns
(bool
) {
functiontransfer
transfer
address
_to
uint256
_value
public
returns
bool
require
(_to !=
address
(0
));
require
(_to !=
address
(0
));
require
!=
address
0
require
(_value <=
balances[msg
.sender
]);
require
(_value <=
balances[msg
.sender
]);
require
<=
msg
sender
balances[msg
.sender
] =
balances[msg
.sender
] –
_value;
balances[msg
.sender
] =
balances[msg
.sender
] –
_value;
msg
sender
=
msg
sender
_

balances[_to] =

balances[_to] +

_value;

balances[_to] =

balances[_to] +

_value;

=

+

return

true

;

return

true

;

return

true

}

}

functionbalanceOf

(address

_owner

) public

view

returns

(uint256

) {

functionbalanceOf

(address

_owner

) public

view

returns

(uint256

) {

functionbalanceOf

balanceOf

address

_owner

public

view

returns

uint256

return

balances[_owner];

return

balances[_owner];

return

}

}

}

}

view raw

[

BasicToken.sol

](https://gist.github.com/ajsantander/dce951a95e7608bc29d7f5deeb6e2ecf#file-basictoken-sol)

hosted with ❤ by GitHub

view raw

[

BasicToken.sol

](https://gist.github.com/ajsantander/dce951a95e7608bc29d7f5deeb6e2ecf#file-basictoken-sol)

GitHubNote: This contract is susceptible to an overflow attack, but we're just keeping it simple here for the purpose at hand.

Note: This contract is susceptible to an overflow attack, but we're just keeping it simple here for the purpose at hand.

overflow attack#### Compiling the contract

To compile the contract, we'll be using Remix. Go ahead and create a new contract by clicking on the + button on the top left, above the file browser area. Set the filename to BasicToken.sol. Now, paste the above code into the editor section.

RemixIn the right-hand section, go to the Settings

tab and make sure Enable Optimization

is selected

. Also, verify that the selected version of the Solidity compiler is "version:0.4.24+commit.e67f0147.Emscripten.clang

". These two details are very important, otherwise you'll be looking at slightly different bytecode from what will be discussed here.

Settings

make sure Enable Optimization

is selected

Enable Optimization

version:0.4.24+commit.e67f0147.Emscripten.clang

If you go to the Compile

tab and click on the Details

button, you should see a popup with all the stuff that the Solidity compiler generates, one of which is a JSON object named BYTECODE

that has an "object" property, which is the compiled code of the contract. It looks like this:

Compile

Details

BYTECODE

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

608060405234801561001057600080fd5b5060405160208061021783398101604090815290516000818155338152600160205291909120556101d180610046600039600 0f30060806040526004361061 0

view raw

[

BasicToken.evm

](https://gist.github.com/ajsantander/03a4a183756980ef0865825bea96d6f5#file-basictoken-evm)

hosted with ❤ by GitHub

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

608060405234801561001057600080fd5b5060405160208061021783398101604090815290516000818155338152600160205291909120556101d180610046600039600 0f30060806040526004361061 0

view raw

[

BasicToken.evm

](https://gist.github.com/ajsantander/03a4a183756980ef0865825bea96d6f5#file-basictoken-evm)

hosted with ❤ by GitHub

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

608060405234801561001057600080fd5b5060405160208061021783398101604090815290516000818155338152600160205291909120556101d180610046600039600 0f30060806040526004361061 0

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

608060405234801561001057600080fd5b5060405160208061021783398101604090815290516000818155338152600160205291909120556101d180610046600039600 0f30060806040526004361061 0

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

608060405234801561001057600080fd5b5060405160208061021783398101604090815290516000818155338152600160205291909120556101d180610046600039600 0f30060806040526004361061 0

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

6080604052348015610010576000080fd5b506040516020806102178339810160409081529051600081815533815260016020529190912055610 1d18061004660003960000f3006080604052600436106100

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

6080604052348015610010576000080fd5b506040516020806102178339810160409081529051600081815533815260016020529190912055610 1d18061004660003960000f3006080604052600436106100

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

[Show hidden characters

]()

6080604052348015610010576000080fd5b506040516020806102178339810160409081529051600081815533815260016020529190912055610 1d18061004660003960000f3006080604052600436106100

6080604052348015610010576000080fd5b506040516020806102178339810160409081529051600081815533815260016020529190912055610 1d18061004660003960000f3006080604052600436106100

6080604052348015610010576000080fd5b506040516020806102178339810160409081529051600081815533815260016020529190912055610 1d18061004660003960000f3006080604052600436106100

6080604052348015610010576000080fd5b506040516020806102178339810160409081529051600081815533815260016020529190912055610 1d18061004660003960000f3006080604052600436106100

view raw

[

BasicToken.evm

](https://gist.github.com/ajsantander/03a4a183756980ef0865825bea96d6f5#file-basictoken-evm)

hosted with ❤ by GitHub

view raw

[

BasicToken.evm

](https://gist.github.com/ajsantander/03a4a183756980ef0865825bea96d6f5#file-basictoken-evm)

GitHubYup. That's completely unreadable (at least for a normal human being).

Deploying the contract

Next, go to the Run

section in Remix. At the top, make sure you're using the Javascript VM

. This is basically an embedded Javascript EVM + network, our ideal Ethereum playground. Make sure BasicToken is selected in the ComboBox

, and enter the number 10000 in the Deploy input box

. Next, click the Deploy

button. This should deploy an instance of our BasicToken contract, with an initial supply of 10000 tokens owned by the account currently selected at the top of the account ComboBox

, which will hold the totality of our token supply.

Run

Javascript VM

ComboBox

Deploy input box

Deploy

ComboBox

Lower in the Run

tab, in the Deployed Contracts

section, you should see the deployed contract, with fields to interact with its three functions: transfer, balanceOf, and totalSupply. Here, we'll be able to interact with the instance of the contract we just deployed.

Run

Deployed Contracts

But before that, let's take a look at exactly what "deploying" the contract means. At the bottom of the page, in the console area, you should see the log "creation of BasicToken pending…", followed by a transaction entry with various fields: from, to, value, data, logs, and hash. Click on this entry to expand the transaction's info. Even though abbreviated, you should see that the data/input of the transaction is the same bytecode we presented above. This transaction is sent to the 0x0 address, and as a result, a new contract instance is created, with its own address and code. We'll examine this process in detail in the next article.

Disassembling the bytecode

To the right of the transaction's data, still in the console, click on the Debug

button. This will activate the Debugger

tab in Remix's right-hand area. Let's take a look at the Instructions

section. If you scroll down, you should see the following:

Debug

Debugger

Instructions

000

PUSH1

80

002

PUSH1

40

004

MSTORE

005

CALLVALUE

006

DUP1

007

ISZERO

008

PUSH2

0010

011

JUMPI

012

PUSH1

00

014

DUP1

015

REVERT

016

JUMPDEST

017

POP

018

PUSH1

40

020

MLOAD

021

PUSH1

20

023

DUP1

024

PUSH2

0217

027

DUP4

028

CODECOPY

029

DUP2

030

ADD

031

PUSH1

40

033

SWAP1

034

DUP2

035

MSTORE

036

SWAP1

037

MLOAD

038

PUSH1

00

040

DUP2

041

DUP2

042

SSTORE

043

CALLER

044

DUP2

045

MSTORE

046

PUSH1

01

048

PUSH1

20

050

MSTORE

051

SWAP2

052

SWAP1

053

SWAP2

054

SHA3

055

SSTORE

056

PUSH2 01d1

059

DUP1

060

PUSH2

0046

063

PUSH1

00

065

CODECOPY

066

PUSH1

00

068

RETURN

069

STOP

070

PUSH1

80

072

PUSH1

40

074

MSTORE

075

PUSH1

04

077

CALLDATASIZE

078

LT

079

PUSH2

0056

082

JUMPI

083

PUSH4 ffffffff

088

PUSH29

0100000000000000000000000000000000000000000000000000000000

118

PUSH1

00

120

CALLDATALOAD

121

DIV

122

AND

123

PUSH4 18160ddd

128

DUP2

129

EQ

130

PUSH2 005b

133

JUMPI

134

DUP1

135

PUSH4 70a08231

140

EQ

141

PUSH2

0082

144

JUMPI

145

DUP1

146
PUSH4 a9059cbb
151
EQ
152
PUSH2 00b0
155
JUMPI
156
JUMPDEST
157
PUSH1
00
159
DUP1
160
REVERT
161
JUMPDEST
162
CALLVALUE
163
DUP1
164
ISZERO
165
PUSH2
0067
168
JUMPI
169
PUSH1
00
171
DUP1
172
REVERT
173
JUMPDEST
174
POP
175
PUSH2
0070
178
PUSH2 00f5
181
JUMP
182
JUMPDEST
183
PUSH1
40
185
DUP1
186
MLOAD
187
SWAP2
188

DUP3

189

MSTORE

190

MLOAD

191

SWAP1

192

DUP2

193

SWAP1

194

SUB

195

PUSH1

20

197

ADD

198

SWAP1

199

RETURN

200

JUMPDEST

201

CALLVALUE

202

DUP1

203

ISZERO

204

PUSH2 008e

207

JUMPI

208

PUSH1

00

210

DUP1

211

REVERT

212

JUMPDEST

213

POP

214

PUSH2

0070

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

238

PUSH1

04

240

CALLDATALOAD

241

AND

242

PUSH2 00fb

245

JUMP

246

JUMPDEST

247

CALLVALUE

248

DUP1

249

ISZERO

250

PUSH2 00bc

253

JUMPI

254

PUSH1

00

256

DUP1

257

REVERT

258

JUMPDEST

259

POP

260

PUSH2 00e1

263

PUSH20 ffffffffffffffffffffffffffffffffffffffff

284

PUSH1

04

286

CALLDATALOAD

287

AND

288

PUSH1

24

290

CALLDATALOAD

291

PUSH2

0123

294

JUMP

295

JUMPDEST

296

PUSH1

40

298

DUP1

299

MLOAD

300

SWAP2

301

ISZERO

302

ISZERO

303
DUP3
304
MSTORE
305
MLOAD
306
SWAP1
307
DUP2
308
SWAP1
309
SUB
310
PUSH1
20
312
ADD
313
SWAP1
314
RETURN
315
JUMPDEST
316
PUSH1
00
318
SLOAD
319
SWAP1
320
JUMP
321
JUMPDEST
322
PUSH20 ffffffffffffffffffffffffffffffffffffffff
343
AND
344
PUSH1
00
346
SWAP1
347
DUP2
348
MSTORE
349
PUSH1
01
351
PUSH1
20
353
MSTORE
354
PUSH1
40

356

SWAP1

357

SHA3

358

SLOAD

359

SWAP1

360

JUMP

361

JUMPDEST

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

386

AND

387

ISZERO

388

ISZERO

389

PUSH2

0147

392

JUMPI

393

PUSH1

00

395

DUP1

396

REVERT

397

JUMPDEST

398

CALLER

399

PUSH1

00

401

SWAP1

402

DUP2

403

MSTORE

404

PUSH1

01

406

PUSH1

20

408

MSTORE

409

PUSH1

40

411

SWAP1

412

SHA3

413

SLOAD

414

DUP3

415

GT

416

ISZERO

417

PUSH2

0163

420

JUMPI

421

PUSH1

00

423

DUP1

424

REVERT

425

JUMPDEST

426

POP

427

CALLER

428

PUSH1

00

430

SWAP1

431

DUP2

432

MSTORE

433

PUSH1

01

435

PUSH1

20

437

DUP2

438

SWAP1

439

MSTORE

440

PUSH1

40

442

DUP1

443

DUP4

444

SHA3

445

DUP1

446

SLOAD

447

DUP6

448

SWAP1

449

SUB

450

SWAP1

451

SSTORE

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

473

DUP6

474

AND

475

DUP4

476

MSTORE

477

SWAP1

478

SWAP2

479

SHA3

480

DUP1

481

SLOAD

482

DUP4

483

ADD

484

SWAP1

485

SSTORE

486

SWAP3

487

SWAP2

488

POP

489

POP

490

JUMP

491

STOP

492

LOG1

493

PUSH6 627a7a723058

500

SHA3

501

INVALID

502

INVALID

503

SWAP10

504

DELEGATECALL

505

GASLIMIT

506

SWAP7

507

TIMESTAMP

508

DUP8

509

INVALID

510

INVALID

511

INVALID

512

SWAP4

513

LOG4

514

SWAP1

515

JUMPI

516

INVALID

517

CALLVALUE

518

INVALID

519

BLOCKHASH

520

INVALID

521

SWAP2

522

INVALID

523

INVALID

524

INVALID

525

INVALID

526

CALLCODE

527

SWAP13

528

DUP9

529

INVALID

530

INVALID

531

RETURN

532

INVALID

533

STOP

534

INVALID

535

STOP

536

STOP

537

STOP

538

STOP

539

STOP

540

STOP

541

STOP

542

STOP

543

STOP

544

STOP

545

STOP

546

STOP

547

STOP

548

STOP

549

STOP

550

STOP

551

STOP

552

STOP

553

STOP

554

STOP

555

STOP

556

STOP

557

STOP

558

STOP

559

STOP

560

STOP

561

STOP

562

STOP

563

STOP

564

STOP

565

INVALID

566

LT

view raw

[

BasicToken.asm

](https://gist.github.com/ajsantander/60bd8d6f88725663f89a67a7c51672c7#file-basictoken-asm)

hosted with ❤ by GitHub

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

000

PUSH1

80

002

PUSH1

40

004

MSTORE

005

CALLVALUE

006

DUP1

007

ISZERO

008

PUSH2

0010

011

JUMPI

012

PUSH1

00

014

DUP1

015

REVERT

016

JUMPDEST

017

POP

018

PUSH1

40

020

MLOAD

021

PUSH1

20

023
DUP1
024
PUSH2
0217
027
DUP4
028
CODECOPY
029
DUP2
030
ADD
031
PUSH1
40
033
SWAP1
034
DUP2
035
MSTORE
036
SWAP1
037
MLOAD
038
PUSH1
00
040
DUP2
041
DUP2
042
SSTORE
043
CALLER
044
DUP2
045
MSTORE
046
PUSH1
01
048
PUSH1
20
050
MSTORE
051
SWAP2
052
SWAP1
053
SWAP2
054
SHA3
055
SSTORE
056

PUSH2 01d1

059

DUP1

060

PUSH2

0046

063

PUSH1

00

065

CODECOPY

066

PUSH1

00

068

RETURN

069

STOP

070

PUSH1

80

072

PUSH1

40

074

MSTORE

075

PUSH1

04

077

CALLDATASIZE

078

LT

079

PUSH2

0056

082

JUMPI

083

PUSH4 ffffffff

088

PUSH29

0100000000000000000000000000000000000000000000000000000000

118

PUSH1

00

120

CALLDATALOAD

121

DIV

122

AND

123

PUSH4 18160ddd

128

DUP2

129

EQ

130

PUSH2 005b

133

JUMPI

134

DUP1

135

PUSH4 70a08231

140

EQ

141

PUSH2

0082

144

JUMPI

145

DUP1

146

PUSH4 a9059cbb

151

EQ

152

PUSH2 00b0

155

JUMPI

156

JUMPDEST

157

PUSH1

00

159

DUP1

160

REVERT

161

JUMPDEST

162

CALLVALUE

163

DUP1

164

ISZERO

165

PUSH2

0067

168

JUMPI

169

PUSH1

00

171

DUP1

172

REVERT

173

JUMPDEST

174

POP

175

PUSH2

0070

178

PUSH2 00f5

181

JUMP

182

JUMPDEST

183

PUSH1

40

185

DUP1

186

MLOAD

187

SWAP2

188

DUP3

189

MSTORE

190

MLOAD

191

SWAP1

192

DUP2

193

SWAP1

194

SUB

195

PUSH1

20

197

ADD

198

SWAP1

199

RETURN

200

JUMPDEST

201

CALLVALUE

202

DUP1

203

ISZERO

204

PUSH2 008e

207

JUMPI

208

PUSH1

00

210

DUP1

211

REVERT

212

JUMPDEST

213

POP

214

PUSH2

0070

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

238

PUSH1

04

240

CALLDATALOAD

241

AND

242

PUSH2 00fb

245

JUMP

246

JUMPDEST

247

CALLVALUE

248

DUP1

249

ISZERO

250

PUSH2 00bc

253

JUMPI

254

PUSH1

00

256

DUP1

257

REVERT

258

JUMPDEST

259

POP

260

PUSH2 00e1

263

PUSH20 ffffffffffffffffffffffffffffffffffffffff

284

PUSH1

04

286

CALLDATALOAD

287

AND

288

PUSH1

24

290

CALLDATALOAD

291

PUSH2

0123

294

JUMP

295
JUMPDEST
296
PUSH1
40
298
DUP1
299
MLOAD
300
SWAP2
301
ISZERO
302
ISZERO
303
DUP3
304
MSTORE
305
MLOAD
306
SWAP1
307
DUP2
308
SWAP1
309
SUB
310
PUSH1
20
312
ADD
313
SWAP1
314
RETURN
315
JUMPDEST
316
PUSH1
00
318
SLOAD
319
SWAP1
320
JUMP
321
JUMPDEST
322
PUSH20 ffffffffffffffffffffffffffffffffffffffff
343
AND
344
PUSH1
00
346
SWAP1

347

DUP2

348

MSTORE

349

PUSH1

01

351

PUSH1

20

353

MSTORE

354

PUSH1

40

356

SWAP1

357

SHA3

358

SLOAD

359

SWAP1

360

JUMP

361

JUMPDEST

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

386

AND

387

ISZERO

388

ISZERO

389

PUSH2

0147

392

JUMPI

393

PUSH1

00

395

DUP1

396

REVERT

397

JUMPDEST

398

CALLER

399

PUSH1

00

401

SWAP1

402

DUP2

403

MSTORE

404

PUSH1

01

406

PUSH1

20

408

MSTORE

409

PUSH1

40

411

SWAP1

412

SHA3

413

SLOAD

414

DUP3

415

GT

416

ISZERO

417

PUSH2

0163

420

JUMPI

421

PUSH1

00

423

DUP1

424

REVERT

425

JUMPDEST

426

POP

427

CALLER

428

PUSH1

00

430

SWAP1

431

DUP2

432

MSTORE

433

PUSH1

01

435

PUSH1

20

437

DUP2

438

SWAP1

439

MSTORE

440

PUSH1

40

442

DUP1

443

DUP4

444

SHA3

445

DUP1

446

SLOAD

447

DUP6

448

SWAP1

449

SUB

450

SWAP1

451

SSTORE

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

473

DUP6

474

AND

475

DUP4

476

MSTORE

477

SWAP1

478

SWAP2

479

SHA3

480

DUP1

481

SLOAD

482

DUP4

483

ADD

484

SWAP1

485

SSTORE

486

SWAP3

487

SWAP2

488

POP

489

POP

490

JUMP

491

STOP

492

LOG1

493

PUSH6 627a7a723058

500

SHA3

501

INVALID

502

INVALID

503

SWAP10

504

DELEGATECALL

505

GASLIMIT

506

SWAP7

507

TIMESTAMP

508

DUP8

509

INVALID

510

INVALID

511

INVALID

512

SWAP4

513

LOG4

514

SWAP1

515

JUMPI

516

INVALID

517

CALLVALUE

518

INVALID

519

BLOCKHASH

520

INVALID

521

SWAP2

522

INVALID

523
INVALID
524
INVALID
525
INVALID
526
CALLCODE
527
SWAP13
528
DUP9
529
INVALID
530
INVALID
531
RETURN
532
INVALID
533
STOP
534
INVALID
535
STOP
536
STOP
537
STOP
538
STOP
539
STOP
540
STOP
541
STOP
542
STOP
543
STOP
544
STOP
545
STOP
546
STOP
547
STOP
548
STOP
549
STOP
550
STOP
551
STOP
552
STOP

553

STOP

554

STOP

555

STOP

556

STOP

557

STOP

558

STOP

559

STOP

560

STOP

561

STOP

562

STOP

563

STOP

564

STOP

565

INVALID

566

LT

view raw

[

BasicToken.asm

](https://gist.github.com/ajsantander/60bd8d6f88725663f89a67a7c51672c7#file-basictoken-asm)

hosted with ❤ by GitHub

000

PUSH1

80

002

PUSH1

40

004

MSTORE

005

CALLVALUE

006

DUP1

007

ISZERO

008

PUSH2

0010

011

JUMPI

012

PUSH1

00

014

DUP1

015

REVERT

016

JUMPDEST

017

POP

018

PUSH1

40

020

MLOAD

021

PUSH1

20

023

DUP1

024

PUSH2

0217

027

DUP4

028

CODECOPY

029

DUP2

030

ADD

031

PUSH1

40

033

SWAP1

034

DUP2

035

MSTORE

036

SWAP1

037

MLOAD

038

PUSH1

00

040

DUP2

041

DUP2

042

SSTORE

043

CALLER

044

DUP2

045

MSTORE

046

PUSH1

01

048

PUSH1

20

050

MSTORE

051

SWAP2

052

SWAP1

053

SWAP2

054

SHA3

055

SSTORE

056

PUSH2 01d1

059

DUP1

060

PUSH2

0046

063

PUSH1

00

065

CODECOPY

066

PUSH1

00

068

RETURN

069

STOP

070

PUSH1

80

072

PUSH1

40

074

MSTORE

075

PUSH1

04

077

CALLDATASIZE

078

LT

079

PUSH2

0056

082

JUMPI

083

PUSH4 ffffffff

088

PUSH29

0100000000000000000000000000000000000000000000000000000000

118

PUSH1

00

120

CALLDATALOAD

121

DIV

122

AND

123

PUSH4 18160ddd

128

DUP2

129

EQ

130

PUSH2 005b

133

JUMPI

134

DUP1

135

PUSH4 70a08231

140

EQ

141

PUSH2

0082

144

JUMPI

145

DUP1

146

PUSH4 a9059cbb

151

EQ

152

PUSH2 00b0

155

JUMPI

156

JUMPDEST

157

PUSH1

00

159

DUP1

160

REVERT

161

JUMPDEST

162

CALLVALUE

163

DUP1

164

ISZERO

165

PUSH2

0067

168

JUMPI

169
PUSH1
00
171
DUP1
172
REVERT
173
JUMPDEST
174
POP
175
PUSH2
0070
178
PUSH2 00f5
181
JUMP
182
JUMPDEST
183
PUSH1
40
185
DUP1
186
MLOAD
187
SWAP2
188
DUP3
189
MSTORE
190
MLOAD
191
SWAP1
192
DUP2
193
SWAP1
194
SUB
195
PUSH1
20
197
ADD
198
SWAP1
199
RETURN
200
JUMPDEST
201
CALLVALUE
202
DUP1
203
ISZERO

204

PUSH2 008e

207

JUMPI

208

PUSH1

00

210

DUP1

211

REVERT

212

JUMPDEST

213

POP

214

PUSH2

0070

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

238

PUSH1

04

240

CALLDATALOAD

241

AND

242

PUSH2 00fb

245

JUMP

246

JUMPDEST

247

CALLVALUE

248

DUP1

249

ISZERO

250

PUSH2 00bc

253

JUMPI

254

PUSH1

00

256

DUP1

257

REVERT

258

JUMPDEST

259

POP

260

PUSH2 00e1

263

PUSH20 ffffffffffffffffffffffffffffffffffffffff

284

PUSH1

04
286
CALLDATALOAD
287
AND
288
PUSH1
24
290
CALLDATALOAD
291
PUSH2
0123
294
JUMP
295
JUMPDEST
296
PUSH1
40
298
DUP1
299
MLOAD
300
SWAP2
301
ISZERO
302
ISZERO
303
DUP3
304
MSTORE
305
MLOAD
306
SWAP1
307
DUP2
308
SWAP1
309
SUB
310
PUSH1
20
312
ADD
313
SWAP1
314
RETURN
315
JUMPDEST
316
PUSH1
00
318
SLOAD

319

SWAP1

320

JUMP

321

JUMPDEST

322

PUSH20 ffffffffffffffffffffffffffffffffffffffff

343

AND

344

PUSH1

00

346

SWAP1

347

DUP2

348

MSTORE

349

PUSH1

01

351

PUSH1

20

353

MSTORE

354

PUSH1

40

356

SWAP1

357

SHA3

358

SLOAD

359

SWAP1

360

JUMP

361

JUMPDEST

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

386

AND

387

ISZERO

388

ISZERO

389

PUSH2

0147

392

JUMPI

393
PUSH1
00
395
DUP1
396
REVERT
397
JUMPDEST
398
CALLER
399
PUSH1
00
401
SWAP1
402
DUP2
403
MSTORE
404
PUSH1
01
406
PUSH1
20
408
MSTORE
409
PUSH1
40
411
SWAP1
412
SHA3
413
SLOAD
414
DUP3
415
GT
416
ISZERO
417
PUSH2
0163
420
JUMPI
421
PUSH1
00
423
DUP1
424
REVERT
425
JUMPDEST
426
POP
427

CALLER

428

PUSH1

00

430

SWAP1

431

DUP2

432

MSTORE

433

PUSH1

01

435

PUSH1

20

437

DUP2

438

SWAP1

439

MSTORE

440

PUSH1

40

442

DUP1

443

DUP4

444

SHA3

445

DUP1

446

SLOAD

447

DUP6

448

SWAP1

449

SUB

450

SWAP1

451

SSTORE

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

473

DUP6

474

AND

475

DUP4

476

MSTORE

477

SWAP1

478

SWAP2

479

SHA3

480

DUP1

481

SLOAD

482

DUP4

483

ADD

484

SWAP1

485

SSTORE

486

SWAP3

487

SWAP2

488

POP

489

POP

490

JUMP

491

STOP

492

LOG1

493

PUSH6 627a7a723058

500

SHA3

501

INVALID

502

INVALID

503

SWAP10

504

DELEGATECALL

505

GASLIMIT

506

SWAP7

507

TIMESTAMP

508

DUP8

509

INVALID

510

INVALID

511

INVALID

512

SWAP4

513

LOG4

514

SWAP1

515

JUMPI

516

INVALID

517

CALLVALUE

518

INVALID

519

BLOCKHASH

520

INVALID

521

SWAP2

522

INVALID

523

INVALID

524

INVALID

525

INVALID

526

CALLCODE

527

SWAP13

528

DUP9

529

INVALID

530

INVALID

531

RETURN

532

INVALID

533

STOP

534

INVALID

535

STOP

536

STOP

537

STOP

538

STOP

539

STOP

540

STOP

541

STOP

542

STOP

543

STOP

544

STOP

545

STOP

546

STOP

547

STOP

548

STOP

549

STOP

550

STOP

551

STOP

552

STOP

553

STOP

554

STOP

555

STOP

556

STOP

557

STOP

558

STOP

559

STOP

560

STOP

561

STOP

562

STOP

563

STOP

564

STOP

565

INVALID

566

LT

000

PUSH1

80

002

PUSH1

40

004

MSTORE

005

CALLVALUE

006

DUP1

007

ISZERO

008

PUSH2

0010

011

JUMPI

012

PUSH1

00

014

DUP1

015

REVERT

016

JUMPDEST

017

POP

018

PUSH1

40

020

MLOAD

021

PUSH1

20

023

DUP1

024

PUSH2

0217

027

DUP4

028

CODECOPY

029

DUP2

030

ADD

031

PUSH1

40

033

SWAP1

034

DUP2

035

MSTORE

036

SWAP1

037

MLOAD

038

PUSH1

00

040

DUP2

041

DUP2

042

SSTORE

043
CALLER
044
DUP2
045
MSTORE
046
PUSH1
01
048
PUSH1
20
050
MSTORE
051
SWAP2
052
SWAP1
053
SWAP2
054
SHA3
055
SSTORE
056
PUSH2 01d1
059
DUP1
060
PUSH2
0046
063
PUSH1
00
065
CODECOPY
066
PUSH1
00
068
RETURN
069
STOP
070
PUSH1
80
072
PUSH1
40
074
MSTORE
075
PUSH1
04
077
CALLDATASIZE
078
LT
079
PUSH2

0056

082

JUMPI

083

PUSH4 ffffffff

088

PUSH29

0100000000000000000000000000000000000000000000000000000000

118

PUSH1

00

120

CALLDATALOAD

121

DIV

122

AND

123

PUSH4 18160ddd

128

DUP2

129

EQ

130

PUSH2 005b

133

JUMPI

134

DUP1

135

PUSH4 70a08231

140

EQ

141

PUSH2

0082

144

JUMPI

145

DUP1

146

PUSH4 a9059cbb

151

EQ

152

PUSH2 00b0

155

JUMPI

156

JUMPDEST

157

PUSH1

00

159

DUP1

160

REVERT

161

JUMPDEST

162

CALLVALUE

163

DUP1

164

ISZERO

165

PUSH2

0067

168

JUMPI

169

PUSH1

00

171

DUP1

172

REVERT

173

JUMPDEST

174

POP

175

PUSH2

0070

178

PUSH2 00f5

181

JUMP

182

JUMPDEST

183

PUSH1

40

185

DUP1

186

MLOAD

187

SWAP2

188

DUP3

189

MSTORE

190

MLOAD

191

SWAP1

192

DUP2

193

SWAP1

194

SUB

195

PUSH1

20

197

ADD

198

SWAP1

199
RETURN
200
JUMPDEST
201
CALLVALUE
202
DUP1
203
ISZERO
204
PUSH2 008e
207
JUMPI
208
PUSH1
00
210
DUP1
211
REVERT
212
JUMPDEST
213
POP
214
PUSH2
0070
217
PUSH20 ffffffffffffffffffffffffffffffffffffffff
238
PUSH1
04
240
CALLDATALOAD
241
AND
242
PUSH2 00fb
245
JUMP
246
JUMPDEST
247
CALLVALUE
248
DUP1
249
ISZERO
250
PUSH2 00bc
253
JUMPI
254
PUSH1
00
256
DUP1
257
REVERT

258

JUMPDEST

259

POP

260

PUSH2 00e1

263

PUSH20 ffffffffffffffffffffffffffffffffffffffff

284

PUSH1

04

286

CALLDATALOAD

287

AND

288

PUSH1

24

290

CALLDATALOAD

291

PUSH2

0123

294

JUMP

295

JUMPDEST

296

PUSH1

40

298

DUP1

299

MLOAD

300

SWAP2

301

ISZERO

302

ISZERO

303

DUP3

304

MSTORE

305

MLOAD

306

SWAP1

307

DUP2

308

SWAP1

309

SUB

310

PUSH1

20

312

ADD

313

SWAP1

314

RETURN

315

JUMPDEST

316

PUSH1

00

318

SLOAD

319

SWAP1

320

JUMP

321

JUMPDEST

322

PUSH20 ffffffffffffffffffffffffffffffffffffffff

343

AND

344

PUSH1

00

346

SWAP1

347

DUP2

348

MSTORE

349

PUSH1

01

351

PUSH1

20

353

MSTORE

354

PUSH1

40

356

SWAP1

357

SHA3

358

SLOAD

359

SWAP1

360

JUMP

361

JUMPDEST

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

386

AND
387
ISZERO
388
ISZERO
389
PUSH2
0147
392
JUMPI
393
PUSH1
00
395
DUP1
396
REVERT
397
JUMPDEST
398
CALLER
399
PUSH1
00
401
SWAP1
402
DUP2
403
MSTORE
404
PUSH1
01
406
PUSH1
20
408
MSTORE
409
PUSH1
40
411
SWAP1
412
SHA3
413
SLOAD
414
DUP3
415
GT
416
ISZERO
417
PUSH2
0163
420
JUMPI
421
PUSH1

00

423

DUP1

424

REVERT

425

JUMPDEST

426

POP

427

CALLER

428

PUSH1

00

430

SWAP1

431

DUP2

432

MSTORE

433

PUSH1

01

435

PUSH1

20

437

DUP2

438

SWAP1

439

MSTORE

440

PUSH1

40

442

DUP1

443

DUP4

444

SHA3

445

DUP1

446

SLOAD

447

DUP6

448

SWAP1

449

SUB

450

SWAP1

451

SSTORE

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

473

DUP6

474

AND
475
DUP4
476
MSTORE
477
SWAP1
478
SWAP2
479
SHA3
480
DUP1
481
SLOAD
482
DUP4
483
ADD
484
SWAP1
485
SSTORE
486
SWAP3
487
SWAP2
488
POP
489
POP
490
JUMP
491
STOP
492
LOG1
493
PUSH6 627a7a723058
500
SHA3
501
INVALID
502
INVALID
503
SWAP10
504
DELEGATECALL
505
GASLIMIT
506
SWAP7
507
TIMESTAMP
508
DUP8
509
INVALID
510

INVALID
511
INVALID
512
SWAP4
513
LOG4
514
SWAP1
515
JUMPI
516
INVALID
517
CALLVALUE
518
INVALID
519
BLOCKHASH
520
INVALID
521
SWAP2
522
INVALID
523
INVALID
524
INVALID
525
INVALID
526
CALLCODE
527
SWAP13
528
DUP9
529
INVALID
530
INVALID
531
RETURN
532
INVALID
533
STOP
534
INVALID
535
STOP
536
STOP
537
STOP
538
STOP
539
STOP
540

STOP

541

STOP

542

STOP

543

STOP

544

STOP

545

STOP

546

STOP

547

STOP

548

STOP

549

STOP

550

STOP

551

STOP

552

STOP

553

STOP

554

STOP

555

STOP

556

STOP

557

STOP

558

STOP

559

STOP

560

STOP

561

STOP

562

STOP

563

STOP

564

STOP

565

INVALID

566

LT

[Show hidden characters

]()

000

PUSH1

80

002
PUSH1
40
004
MSTORE
005
CALLVALUE
006
DUP1
007
ISZERO
008
PUSH2
0010
011
JUMPI
012
PUSH1
00
014
DUP1
015
REVERT
016
JUMPDEST
017
POP
018
PUSH1
40
020
MLOAD
021
PUSH1
20
023
DUP1
024
PUSH2
0217
027
DUP4
028
CODECOPY
029
DUP2
030
ADD
031
PUSH1
40
033
SWAP1
034
DUP2
035
MSTORE
036
SWAP1
037

MLOAD

038

PUSH1

00

040

DUP2

041

DUP2

042

SSTORE

043

CALLER

044

DUP2

045

MSTORE

046

PUSH1

01

048

PUSH1

20

050

MSTORE

051

SWAP2

052

SWAP1

053

SWAP2

054

SHA3

055

SSTORE

056

PUSH2 01d1

059

DUP1

060

PUSH2

0046

063

PUSH1

00

065

CODECOPY

066

PUSH1

00

068

RETURN

069

STOP

070

PUSH1

80

072

PUSH1

40

074

MSTORE

075

PUSH1

04

077

CALLDATASIZE

078

LT

079

PUSH2

0056

082

JUMPI

083

PUSH4 ffffffff

088

PUSH29

0100000000000000000000000000000000000000000000000000000000

118

PUSH1

00

120

CALLDATALOAD

121

DIV

122

AND

123

PUSH4 18160ddd

128

DUP2

129

EQ

130

PUSH2 005b

133

JUMPI

134

DUP1

135

PUSH4 70a08231

140

EQ

141

PUSH2

0082

144

JUMPI

145

DUP1

146

PUSH4 a9059cbb

151

EQ

152

PUSH2 00b0

155

JUMPI

156

JUMPDEST

157

PUSH1

00

159

DUP1

160

REVERT

161

JUMPDEST

162

CALLVALUE

163

DUP1

164

ISZERO

165

PUSH2

0067

168

JUMPI

169

PUSH1

00

171

DUP1

172

REVERT

173

JUMPDEST

174

POP

175

PUSH2

0070

178

PUSH2 00f5

181

JUMP

182

JUMPDEST

183

PUSH1

40

185

DUP1

186

MLOAD

187

SWAP2

188

DUP3

189

MSTORE

190

MLOAD

191

SWAP1

192

DUP2

193

SWAP1

194

SUB

195

PUSH1

20

197

ADD

198

SWAP1

199

RETURN

200

JUMPDEST

201

CALLVALUE

202

DUP1

203

ISZERO

204

PUSH2 008e

207

JUMPI

208

PUSH1

00

210

DUP1

211

REVERT

212

JUMPDEST

213

POP

214

PUSH2

0070

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

238

PUSH1

04

240

CALLDATALOAD

241

AND

242

PUSH2 00fb

245

JUMP

246

JUMPDEST

247

CALLVALUE

248

DUP1

249

ISZERO

250

PUSH2 00bc

253

JUMPI

254

PUSH1

00

256

DUP1

257

REVERT

258

JUMPDEST

259

POP

260

PUSH2 00e1

263

PUSH20 ffffffffffffffffffffffffffffffffffffffff

284

PUSH1

04

286

CALLDATALOAD

287

AND

288

PUSH1

24

290

CALLDATALOAD

291

PUSH2

0123

294

JUMP

295

JUMPDEST

296

PUSH1

40

298

DUP1

299

MLOAD

300

SWAP2

301

ISZERO

302

ISZERO

303

DUP3

304

MSTORE

305

MLOAD

306

SWAP1

307

DUP2

308
SWAP1
309
SUB
310
PUSH1
20
312
ADD
313
SWAP1
314
RETURN
315
JUMPDEST
316
PUSH1
00
318
SLOAD
319
SWAP1
320
JUMP
321
JUMPDEST
322
PUSH20 ffffffffffffffffffffffffffffffffffffffff
343
AND
344
PUSH1
00
346
SWAP1
347
DUP2
348
MSTORE
349
PUSH1
01
351
PUSH1
20
353
MSTORE
354
PUSH1
40
356
SWAP1
357
SHA3
358
SLOAD
359
SWAP1
360
JUMP

361

JUMPDEST

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

386

AND

387

ISZERO

388

ISZERO

389

PUSH2

0147

392

JUMPI

393

PUSH1

00

395

DUP1

396

REVERT

397

JUMPDEST

398

CALLER

399

PUSH1

00

401

SWAP1

402

DUP2

403

MSTORE

404

PUSH1

01

406

PUSH1

20

408

MSTORE

409

PUSH1

40

411

SWAP1

412

SHA3

413

SLOAD

414

DUP3

415

GT

416

ISZERO

417

PUSH2

0163

420

JUMPI

421

PUSH1

00

423

DUP1

424

REVERT

425

JUMPDEST

426

POP

427

CALLER

428

PUSH1

00

430

SWAP1

431

DUP2

432

MSTORE

433

PUSH1

01

435

PUSH1

20

437

DUP2

438

SWAP1

439

MSTORE

440

PUSH1

40

442

DUP1

443

DUP4

444

SHA3

445

DUP1

446

SLOAD

447

DUP6

448

SWAP1

449

SUB

450

SWAP1

451

SSTORE

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

473

DUP6

474

AND

475

DUP4

476

MSTORE

477

SWAP1

478

SWAP2

479

SHA3

480

DUP1

481

SLOAD

482

DUP4

483

ADD

484

SWAP1

485

SSTORE

486

SWAP3

487

SWAP2

488

POP

489

POP

490

JUMP

491

STOP

492

LOG1

493

PUSH6 627a7a723058

500

SHA3

501

INVALID

502

INVALID

503

SWAP10

504

DELEGATECALL

505

GASLIMIT
506
SWAP7
507
TIMESTAMP
508
DUP8
509
INVALID
510
INVALID
511
INVALID
512
SWAP4
513
LOG4
514
SWAP1
515
JUMPI
516
INVALID
517
CALLVALUE
518
INVALID
519
BLOCKHASH
520
INVALID
521
SWAP2
522
INVALID
523
INVALID
524
INVALID
525
INVALID
526
CALLCODE
527
SWAP13
528
DUP9
529
INVALID
530
INVALID
531
RETURN
532
INVALID
533
STOP
534
INVALID
535

STOP
536
STOP
537
STOP
538
STOP
539
STOP
540
STOP
541
STOP
542
STOP
543
STOP
544
STOP
545
STOP
546
STOP
547
STOP
548
STOP
549
STOP
550
STOP
551
STOP
552
STOP
553
STOP
554
STOP
555
STOP
556
STOP
557
STOP
558
STOP
559
STOP
560
STOP
561
STOP
562
STOP
563
STOP
564
STOP
565

INVALID

566

LT

This file contains bidirectional Unicode text that may be interpreted or compiled differently than what appears below. To review, open the file in an editor that reveals hidden Unicode characters.

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

000

PUSH1

80

002

PUSH1

40

004

MSTORE

005

CALLVALUE

006

DUP1

007

ISZERO

008

PUSH2

0010

011

JUMPI

012

PUSH1

00

014

DUP1

015

REVERT

016

JUMPDEST

017

POP

018

PUSH1

40

020

MLOAD

021

PUSH1

20

023

DUP1

024

PUSH2

0217

027

DUP4

028

CODECOPY

029

DUP2

030

ADD

031

PUSH1

40

033

SWAP1

034

DUP2

035

MSTORE

036

SWAP1

037

MLOAD

038

PUSH1

00

040

DUP2

041

DUP2

042

SSTORE

043

CALLER

044

DUP2

045

MSTORE

046

PUSH1

01

048

PUSH1

20

050

MSTORE

051

SWAP2

052

SWAP1

053

SWAP2

054

SHA3

055

SSTORE

056

PUSH2 01d1

059

DUP1

060

PUSH2

0046

063

PUSH1

00

065

CODECOPY

066

PUSH1

00

068

RETURN

069

STOP

070

PUSH1

80

072

PUSH1

40

074

MSTORE

075

PUSH1

04

077

CALLDATASIZE

078

LT

079

PUSH2

0056

082

JUMPI

083

PUSH4 ffffffff

088

PUSH29

0100000000000000000000000000000000000000000000000000000000

118

PUSH1

00

120

CALLDATALOAD

121

DIV

122

AND

123

PUSH4 18160ddd

128

DUP2

129

EQ

130

PUSH2 005b

133

JUMPI

134

DUP1

135

PUSH4 70a08231

140

EQ

141

PUSH2

0082

144

JUMPI

145

DUP1

146
PUSH4 a9059cbb
151
EQ
152
PUSH2 00b0
155
JUMPI
156
JUMPDEST
157
PUSH1
00
159
DUP1
160
REVERT
161
JUMPDEST
162
CALLVALUE
163
DUP1
164
ISZERO
165
PUSH2
0067
168
JUMPI
169
PUSH1
00
171
DUP1
172
REVERT
173
JUMPDEST
174
POP
175
PUSH2
0070
178
PUSH2 00f5
181
JUMP
182
JUMPDEST
183
PUSH1
40
185
DUP1
186
MLOAD
187
SWAP2
188

DUP3

189

MSTORE

190

MLOAD

191

SWAP1

192

DUP2

193

SWAP1

194

SUB

195

PUSH1

20

197

ADD

198

SWAP1

199

RETURN

200

JUMPDEST

201

CALLVALUE

202

DUP1

203

ISZERO

204

PUSH2 008e

207

JUMPI

208

PUSH1

00

210

DUP1

211

REVERT

212

JUMPDEST

213

POP

214

PUSH2

0070

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

238

PUSH1

04

240

CALLDATALOAD

241

AND

242

PUSH2 00fb

245

JUMP

246

JUMPDEST

247

CALLVALUE

248

DUP1

249

ISZERO

250

PUSH2 00bc

253

JUMPI

254

PUSH1

00

256

DUP1

257

REVERT

258

JUMPDEST

259

POP

260

PUSH2 00e1

263

PUSH20 ffffffffffffffffffffffffffffffffffffffff

284

PUSH1

04

286

CALLDATALOAD

287

AND

288

PUSH1

24

290

CALLDATALOAD

291

PUSH2

0123

294

JUMP

295

JUMPDEST

296

PUSH1

40

298

DUP1

299

MLOAD

300

SWAP2

301

ISZERO

302

ISZERO

303
DUP3
304
MSTORE
305
MLOAD
306
SWAP1
307
DUP2
308
SWAP1
309
SUB
310
PUSH1
20
312
ADD
313
SWAP1
314
RETURN
315
JUMPDEST
316
PUSH1
00
318
SLOAD
319
SWAP1
320
JUMP
321
JUMPDEST
322
PUSH20 ffffffffffffffffffffffffffffffffffffffff
343
AND
344
PUSH1
00
346
SWAP1
347
DUP2
348
MSTORE
349
PUSH1
01
351
PUSH1
20
353
MSTORE
354
PUSH1
40

356

SWAP1

357

SHA3

358

SLOAD

359

SWAP1

360

JUMP

361

JUMPDEST

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

386

AND

387

ISZERO

388

ISZERO

389

PUSH2

0147

392

JUMPI

393

PUSH1

00

395

DUP1

396

REVERT

397

JUMPDEST

398

CALLER

399

PUSH1

00

401

SWAP1

402

DUP2

403

MSTORE

404

PUSH1

01

406

PUSH1

20

408

MSTORE

409

PUSH1

40

411

SWAP1

412

SHA3

413

SLOAD

414

DUP3

415

GT

416

ISZERO

417

PUSH2

0163

420

JUMPI

421

PUSH1

00

423

DUP1

424

REVERT

425

JUMPDEST

426

POP

427

CALLER

428

PUSH1

00

430

SWAP1

431

DUP2

432

MSTORE

433

PUSH1

01

435

PUSH1

20

437

DUP2

438

SWAP1

439

MSTORE

440

PUSH1

40

442

DUP1

443

DUP4

444

SHA3

445

DUP1

446

SLOAD

447

DUP6

448

SWAP1

449

SUB

450

SWAP1

451

SSTORE

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

473

DUP6

474

AND

475

DUP4

476

MSTORE

477

SWAP1

478

SWAP2

479

SHA3

480

DUP1

481

SLOAD

482

DUP4

483

ADD

484

SWAP1

485

SSTORE

486

SWAP3

487

SWAP2

488

POP

489

POP

490

JUMP

491

STOP

492

LOG1

493

PUSH6 627a7a723058

500

SHA3

501

INVALID

502

INVALID

503

SWAP10

504

DELEGATECALL

505

GASLIMIT

506

SWAP7

507

TIMESTAMP

508

DUP8

509

INVALID

510

INVALID

511

INVALID

512

SWAP4

513

LOG4

514

SWAP1

515

JUMPI

516

INVALID

517

CALLVALUE

518

INVALID

519

BLOCKHASH

520

INVALID

521

SWAP2

522

INVALID

523

INVALID

524

INVALID

525

INVALID

526

CALLCODE

527

SWAP13

528

DUP9

529

INVALID

530

INVALID

531

RETURN

532

INVALID

533

STOP

534

INVALID

535

STOP

536

STOP

537

STOP

538

STOP

539

STOP

540

STOP

541

STOP

542

STOP

543

STOP

544

STOP

545

STOP

546

STOP

547

STOP

548

STOP

549

STOP

550

STOP

551

STOP

552

STOP

553

STOP

554

STOP

555

STOP

556

STOP

557

STOP

558

STOP

559

STOP

560

STOP

561

STOP

562

STOP

563

STOP

564

STOP

565

INVALID

566

LT

000

PUSH1

80

002

PUSH1

40

004

MSTORE

005

CALLVALUE

006

DUP1

007

ISZERO

008

PUSH2

0010

011

JUMPI

012

PUSH1

00

014

DUP1

015

REVERT

016

JUMPDEST

017

POP

018

PUSH1

40

020

MLOAD

021

PUSH1

20

023

DUP1

024

PUSH2

0217

027

DUP4

028

CODECOPY

029

DUP2

030

ADD

031

PUSH1

40

033

SWAP1

034

DUP2

035

MSTORE

036

SWAP1

037

MLOAD

038

PUSH1

00

040

DUP2

041

DUP2

042

SSTORE

043

CALLER

044

DUP2

045

MSTORE

046

PUSH1

01

048

PUSH1

20

050

MSTORE

051

SWAP2

052

SWAP1

053

SWAP2

054

SHA3

055

SSTORE

056

PUSH2 01d1

059

DUP1

060

PUSH2

0046

063

PUSH1

00

065

CODECOPY

066

PUSH1

00

068

RETURN

069

STOP

070

PUSH1

80

072

PUSH1

40

074

MSTORE

075

PUSH1

04

077

CALLDATASIZE

078

LT

079

PUSH2

0056

082

JUMPI

083

PUSH4 ffffffff

088

PUSH29

0100000000000000000000000000000000000000000000000000000000

118

PUSH1

00

120

CALLDATALOAD

121

DIV

122

AND

123

PUSH4 18160ddd

128

DUP2

129

EQ

130

PUSH2 005b

133

JUMPI

134

DUP1

135

PUSH4 70a08231

140

EQ

141

PUSH2

0082

144

JUMPI

145

DUP1

146

PUSH4 a9059cbb

151

EQ

152

PUSH2 00b0

155

JUMPI

156

JUMPDEST

157

PUSH1

00

159

DUP1

160

REVERT

161

JUMPDEST

162

CALLVALUE

163

DUP1

164

ISZERO

165

PUSH2

0067

168

JUMPI

169

PUSH1

00

171

DUP1

172

REVERT

173

JUMPDEST

174

POP

175

PUSH2

0070

178

PUSH2 00f5

181

JUMP

182

JUMPDEST

183

PUSH1

40

185

DUP1

186

MLOAD

187

SWAP2

188

DUP3

189

MSTORE

190

MLOAD

191

SWAP1

192

DUP2

193

SWAP1

194

SUB

195

PUSH1

20

197

ADD

198

SWAP1

199

RETURN

200

JUMPDEST

201

CALLVALUE

202

DUP1

203

ISZERO

204

PUSH2 008e

207

JUMPI

208

PUSH1

00

210

DUP1

211

REVERT

212

JUMPDEST

213

POP

214

PUSH2

0070

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

238

PUSH1

04

240

CALLDATALOAD

241

AND

242

PUSH2 00fb

245

JUMP

246

JUMPDEST

247

CALLVALUE

248

DUP1

249

ISZERO

250

PUSH2 00bc

253

JUMPI

254

PUSH1

00

256

DUP1

257

REVERT

258

JUMPDEST

259

POP

260

PUSH2 00e1

263

PUSH20 ffffffffffffffffffffffffffffffffffffffff

284

PUSH1

04

286

CALLDATALOAD

287

AND

288

PUSH1

24

290

CALLDATALOAD

291

PUSH2

0123

294

JUMP

295

JUMPDEST

296

PUSH1

40

298

DUP1

299

MLOAD

300

SWAP2

301

ISZERO

302

ISZERO

303

DUP3

304

MSTORE

305

MLOAD

306

SWAP1

307

DUP2

308

SWAP1

309

SUB

310

PUSH1

20

312

ADD

313

SWAP1

314

RETURN

315

JUMPDEST

316

PUSH1

00

318

SLOAD

319

SWAP1

320

JUMP

321

JUMPDEST

322

PUSH20 ffffffffffffffffffffffffffffffffffffffff

343

AND

344

PUSH1

00

346

SWAP1

347

DUP2

348

MSTORE

349

PUSH1

01

351

PUSH1

20

353

MSTORE

354

PUSH1

40

356

SWAP1

357

SHA3

358

SLOAD

359

SWAP1

360

JUMP

361

JUMPDEST

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

386

AND

387

ISZERO

388

ISZERO

389

PUSH2

0147

392

JUMPI

393

PUSH1

00

395

DUP1

396

REVERT

397

JUMPDEST

398

CALLER

399

PUSH1

00

401

SWAP1

402

DUP2

403

MSTORE

404
PUSH1
01
406
PUSH1
20
408
MSTORE
409
PUSH1
40
411
SWAP1
412
SHA3
413
SLOAD
414
DUP3
415
GT
416
ISZERO
417
PUSH2
0163
420
JUMPI
421
PUSH1
00
423
DUP1
424
REVERT
425
JUMPDEST
426
POP
427
CALLER
428
PUSH1
00
430
SWAP1
431
DUP2
432
MSTORE
433
PUSH1
01
435
PUSH1
20
437
DUP2
438
SWAP1

439

MSTORE

440

PUSH1

40

442

DUP1

443

DUP4

444

SHA3

445

DUP1

446

SLOAD

447

DUP6

448

SWAP1

449

SUB

450

SWAP1

451

SSTORE

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

473

DUP6

474

AND

475

DUP4

476

MSTORE

477

SWAP1

478

SWAP2

479

SHA3

480

DUP1

481

SLOAD

482

DUP4

483

ADD

484

SWAP1

485

SSTORE

486

SWAP3

487

SWAP2

488

POP

489

POP

490

JUMP

491

STOP

492

LOG1

493

PUSH6 627a7a723058

500

SHA3

501

INVALID

502

INVALID

503

SWAP10

504

DELEGATECALL

505

GASLIMIT

506

SWAP7

507

TIMESTAMP

508

DUP8

509

INVALID

510

INVALID

511

INVALID

512

SWAP4

513

LOG4

514

SWAP1

515

JUMPI

516

INVALID

517

CALLVALUE

518

INVALID

519

BLOCKHASH

520

INVALID

521

SWAP2

522

INVALID

523

INVALID

524

INVALID

525

INVALID
526
CALLCODE
527
SWAP13
528
DUP9
529
INVALID
530
INVALID
531
RETURN
532
INVALID
533
STOP
534
INVALID
535
STOP
536
STOP
537
STOP
538
STOP
539
STOP
540
STOP
541
STOP
542
STOP
543
STOP
544
STOP
545
STOP
546
STOP
547
STOP
548
STOP
549
STOP
550
STOP
551
STOP
552
STOP
553
STOP
554
STOP
555

STOP

556

STOP

557

STOP

558

STOP

559

STOP

560

STOP

561

STOP

562

STOP

563

STOP

564

STOP

565

INVALID

566

LT

Learn more about bidirectional Unicode characters

[Show hidden characters

]()

[Show hidden characters

]()

000

PUSH1

80

002

PUSH1

40

004

MSTORE

005

CALLVALUE

006

DUP1

007

ISZERO

008

PUSH2

0010

011

JUMPI

012

PUSH1

00

014

DUP1

015

REVERT

016
JUMPDEST
017
POP
018
PUSH1
40
020
MLOAD
021
PUSH1
20
023
DUP1
024
PUSH2
0217
027
DUP4
028
CODECOPY
029
DUP2
030
ADD
031
PUSH1
40
033
SWAP1
034
DUP2
035
MSTORE
036
SWAP1
037
MLOAD
038
PUSH1
00
040
DUP2
041
DUP2
042
SSTORE
043
CALLER
044
DUP2
045
MSTORE
046
PUSH1
01
048
PUSH1
20
050

MSTORE

051

SWAP2

052

SWAP1

053

SWAP2

054

SHA3

055

SSTORE

056

PUSH2 01d1

059

DUP1

060

PUSH2

0046

063

PUSH1

00

065

CODECOPY

066

PUSH1

00

068

RETURN

069

STOP

070

PUSH1

80

072

PUSH1

40

074

MSTORE

075

PUSH1

04

077

CALLDATASIZE

078

LT

079

PUSH2

0056

082

JUMPI

083

PUSH4 ffffffff

088

PUSH29

0100000000000000000000000000000000000000000000000000000000

118

PUSH1

00

120

CALLDATALOAD

121

DIV

122

AND

123

PUSH4 18160ddd

128

DUP2

129

EQ

130

PUSH2 005b

133

JUMPI

134

DUP1

135

PUSH4 70a08231

140

EQ

141

PUSH2

0082

144

JUMPI

145

DUP1

146

PUSH4 a9059cbb

151

EQ

152

PUSH2 00b0

155

JUMPI

156

JUMPDEST

157

PUSH1

00

159

DUP1

160

REVERT

161

JUMPDEST

162

CALLVALUE

163

DUP1

164

ISZERO

165

PUSH2

0067

168

JUMPI

169

PUSH1

00

171

DUP1

172

REVERT

173

JUMPDEST

174

POP

175

PUSH2

0070

178

PUSH2 00f5

181

JUMP

182

JUMPDEST

183

PUSH1

40

185

DUP1

186

MLOAD

187

SWAP2

188

DUP3

189

MSTORE

190

MLOAD

191

SWAP1

192

DUP2

193

SWAP1

194

SUB

195

PUSH1

20

197

ADD

198

SWAP1

199

RETURN

200

JUMPDEST

201

CALLVALUE

202

DUP1

203

ISZERO

204

PUSH2 008e

207

JUMPI

208

PUSH1

00

210

DUP1

211

REVERT

212

JUMPDEST

213

POP

214

PUSH2

0070

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

238

PUSH1

04

240

CALLDATALOAD

241

AND

242

PUSH2 00fb

245

JUMP

246

JUMPDEST

247

CALLVALUE

248

DUP1

249

ISZERO

250

PUSH2 00bc

253

JUMPI

254

PUSH1

00

256

DUP1

257

REVERT

258

JUMPDEST

259

POP

260

PUSH2 00e1

263

PUSH20 ffffffffffffffffffffffffffffffffffffffff

284

PUSH1

04

286

CALLDATALOAD

287
AND
288
PUSH1
24
290
CALLDATALOAD
291
PUSH2
0123
294
JUMP
295
JUMPDEST
296
PUSH1
40
298
DUP1
299
MLOAD
300
SWAP2
301
ISZERO
302
ISZERO
303
DUP3
304
MSTORE
305
MLOAD
306
SWAP1
307
DUP2
308
SWAP1
309
SUB
310
PUSH1
20
312
ADD
313
SWAP1
314
RETURN
315
JUMPDEST
316
PUSH1
00
318
SLOAD
319
SWAP1
320

JUMP

321

JUMPDEST

322

PUSH20 ffffffffffffffffffffffffffffffffffffffff

343

AND

344

PUSH1

00

346

SWAP1

347

DUP2

348

MSTORE

349

PUSH1

01

351

PUSH1

20

353

MSTORE

354

PUSH1

40

356

SWAP1

357

SHA3

358

SLOAD

359

SWAP1

360

JUMP

361

JUMPDEST

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

386

AND

387

ISZERO

388

ISZERO

389

PUSH2

0147

392

JUMPI

393

PUSH1

00

395
DUP1
396
REVERT
397
JUMPDEST
398
CALLER
399
PUSH1
00
401
SWAP1
402
DUP2
403
MSTORE
404
PUSH1
01
406
PUSH1
20
408
MSTORE
409
PUSH1
40
411
SWAP1
412
SHA3
413
SLOAD
414
DUP3
415
GT
416
ISZERO
417
PUSH2
0163
420
JUMPI
421
PUSH1
00
423
DUP1
424
REVERT
425
JUMPDEST
426
POP
427
CALLER
428
PUSH1

00

430

SWAP1

431

DUP2

432

MSTORE

433

PUSH1

01

435

PUSH1

20

437

DUP2

438

SWAP1

439

MSTORE

440

PUSH1

40

442

DUP1

443

DUP4

444

SHA3

445

DUP1

446

SLOAD

447

DUP6

448

SWAP1

449

SUB

450

SWAP1

451

SSTORE

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

473

DUP6

474

AND

475

DUP4

476

MSTORE

477

SWAP1

478

SWAP2

479

SHA3

480

DUP1

481

SLOAD

482

DUP4

483

ADD

484

SWAP1

485

SSTORE

486

SWAP3

487

SWAP2

488

POP

489

POP

490

JUMP

491

STOP

492

LOG1

493

PUSH6 627a7a723058

500

SHA3

501

INVALID

502

INVALID

503

SWAP10

504

DELEGATECALL

505

GASLIMIT

506

SWAP7

507

TIMESTAMP

508

DUP8

509

INVALID

510

INVALID

511

INVALID

512

SWAP4

513

LOG4

514

SWAP1

515

JUMPI

516

INVALID

517
CALLVALUE
518
INVALID
519
BLOCKHASH
520
INVALID
521
SWAP2
522
INVALID
523
INVALID
524
INVALID
525
INVALID
526
CALLCODE
527
SWAP13
528
DUP9
529
INVALID
530
INVALID
531
RETURN
532
INVALID
533
STOP
534
INVALID
535
STOP
536
STOP
537
STOP
538
STOP
539
STOP
540
STOP
541
STOP
542
STOP
543
STOP
544
STOP
545
STOP
546
STOP

547
STOP
548
STOP
549
STOP
550
STOP
551
STOP
552
STOP
553
STOP
554
STOP
555
STOP
556
STOP
557
STOP
558
STOP
559
STOP
560
STOP
561
STOP
562
STOP
563
STOP
564
STOP
565
INVALID
566
LT
000
PUSH1
80
002
PUSH1
40
004
MSTORE
005
CALLVALUE
006
DUP1
007
ISZERO
008
PUSH2
0010
011
JUMPI
012

PUSH1

00

014

DUP1

015

REVERT

016

JUMPDEST

017

POP

018

PUSH1

40

020

MLOAD

021

PUSH1

20

023

DUP1

024

PUSH2

0217

027

DUP4

028

CODECOPY

029

DUP2

030

ADD

031

PUSH1

40

033

SWAP1

034

DUP2

035

MSTORE

036

SWAP1

037

MLOAD

038

PUSH1

00

040

DUP2

041

DUP2

042

SSTORE

043

CALLER

044

DUP2

045

MSTORE

046

PUSH1

01

048

PUSH1

20

050

MSTORE

051

SWAP2

052

SWAP1

053

SWAP2

054

SHA3

055

SSTORE

056

PUSH2 01d1

059

DUP1

060

PUSH2

0046

063

PUSH1

00

065

CODECOPY

066

PUSH1

00

068

RETURN

069

STOP

070

PUSH1

80

072

PUSH1

40

074

MSTORE

075

PUSH1

04

077

CALLDATASIZE

078

LT

079

PUSH2

0056

082

JUMPI

083

PUSH4 ffffffff

088

PUSH29

0100000000000000000000000000000000000000000000000000000000000000

118

PUSH1

00

120

CALLDATALOAD

121

DIV

122

AND

123

PUSH4 18160ddd

128

DUP2

129

EQ

130

PUSH2 005b

133

JUMPI

134

DUP1

135

PUSH4 70a08231

140

EQ

141

PUSH2

0082

144

JUMPI

145

DUP1

146

PUSH4 a9059cbb

151

EQ

152

PUSH2 00b0

155

JUMPI

156

JUMPDEST

157

PUSH1

00

159

DUP1

160

REVERT

161

JUMPDEST

162

CALLVALUE

163

DUP1

164

ISZERO

165

PUSH2

0067

168

JUMPI

169

PUSH1

00

171

DUP1

172

REVERT

173

JUMPDEST

174

POP

175

PUSH2

0070

178

PUSH2 00f5

181

JUMP

182

JUMPDEST

183

PUSH1

40

185

DUP1

186

MLOAD

187

SWAP2

188

DUP3

189

MSTORE

190

MLOAD

191

SWAP1

192

DUP2

193

SWAP1

194

SUB

195

PUSH1

20

197

ADD

198

SWAP1

199

RETURN

200

JUMPDEST

201

CALLVALUE

202

DUP1

203

ISZERO

204

PUSH2 008e

207

JUMPI

208

PUSH1

00

210

DUP1

211

REVERT

212

JUMPDEST

213

POP

214

PUSH2

0070

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

238

PUSH1

04

240

CALLDATALOAD

241

AND

242

PUSH2 00fb

245

JUMP

246

JUMPDEST

247

CALLVALUE

248

DUP1

249

ISZERO

250

PUSH2 00bc

253

JUMPI

254

PUSH1

00

256

DUP1

257

REVERT

258

JUMPDEST

259

POP

260

PUSH2 00e1

263

PUSH20 ffffffffffffffffffffffffffffffffffffffff

284

PUSH1

04

286

CALLDATALOAD

287

AND

288

PUSH1

24

290

CALLDATALOAD

291

PUSH2

0123

294

JUMP

295

JUMPDEST

296

PUSH1

40

298

DUP1

299

MLOAD

300

SWAP2

301

ISZERO

302

ISZERO

303

DUP3

304

MSTORE

305

MLOAD

306

SWAP1

307

DUP2

308

SWAP1

309

SUB

310

PUSH1

20

312

ADD

313

SWAP1

314

RETURN

315

JUMPDEST

316

PUSH1

00

318

SLOAD

319

SWAP1

320

JUMP

321

JUMPDEST

322

PUSH20 ffffffffffffffffffffffffffffffffffffffff

343

AND

344

PUSH1

00

346

SWAP1

347

DUP2

348

MSTORE

349

PUSH1

01

351

PUSH1

20

353

MSTORE

354

PUSH1

40

356

SWAP1

357

SHA3

358

SLOAD

359

SWAP1

360

JUMP

361

JUMPDEST

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

386

AND

387

ISZERO

388

ISZERO

389

PUSH2

0147

392

JUMPI

393

PUSH1

00

395

DUP1

396

REVERT

397

JUMPDEST

398

CALLER

399

PUSH1

00

401

SWAP1

402

DUP2

403

MSTORE

404

PUSH1

01

406

PUSH1

20

408

MSTORE

409

PUSH1

40

411

SWAP1

412

SHA3

413

SLOAD

414

DUP3

415

GT

416

ISZERO

417

PUSH2

0163

420

JUMPI

421

PUSH1

00

423

DUP1

424

REVERT

425

JUMPDEST

426
POP
427
CALLER
428
PUSH1
00
430
SWAP1
431
DUP2
432
MSTORE
433
PUSH1
01
435
PUSH1
20
437
DUP2
438
SWAP1
439
MSTORE
440
PUSH1
40
442
DUP1
443
DUP4
444
SHA3
445
DUP1
446
SLOAD
447
DUP6
448
SWAP1
449
SUB
450
SWAP1
451
SSTORE
452
PUSH20 ffffffffffffffffffffffffffffffffffffffff
473
DUP6
474
AND
475
DUP4
476
MSTORE
477
SWAP1

478
SWAP2
479
SHA3
480
DUP1
481
SLOAD
482
DUP4
483
ADD
484
SWAP1
485
SSTORE
486
SWAP3
487
SWAP2
488
POP
489
POP
490
JUMP
491
STOP
492
LOG1
493
PUSH6 627a7a723058
500
SHA3
501
INVALID
502
INVALID
503
SWAP10
504
DELEGATECALL
505
GASLIMIT
506
SWAP7
507
TIMESTAMP
508
DUP8
509
INVALID
510
INVALID
511
INVALID
512
SWAP4
513
LOG4

514
SWAP1
515
JUMPI
516
INVALID
517
CALLVALUE
518
INVALID
519
BLOCKHASH
520
INVALID
521
SWAP2
522
INVALID
523
INVALID
524
INVALID
525
INVALID
526
CALLCODE
527
SWAP13
528
DUP9
529
INVALID
530
INVALID
531
RETURN
532
INVALID
533
STOP
534
INVALID
535
STOP
536
STOP
537
STOP
538
STOP
539
STOP
540
STOP
541
STOP
542
STOP
543
STOP

544
STOP
545
STOP
546
STOP
547
STOP
548
STOP
549
STOP
550
STOP
551
STOP
552
STOP
553
STOP
554
STOP
555
STOP
556
STOP
557
STOP
558
STOP
559
STOP
560
STOP
561
STOP
562
STOP
563
STOP
564
STOP
565
INVALID
566
LT
000
PUSH1
80
000
PUSH1
80
000
PUSH1
80
002
PUSH1
40
002
PUSH1

40
002
PUSH1
40
004
MSTORE
004
MSTORE
004
MSTORE
005
CALLVALUE
005
CALLVALUE
005
CALLVALUE
006
DUP1
006
DUP1
006
DUP1
007
ISZERO
007
ISZERO
007
ISZERO
008
PUSH2
0010
008
PUSH2
0010
008
PUSH2
0010
011
JUMPI
011
JUMPI
011
JUMPI
012
PUSH1
00
012
PUSH1
00
012
PUSH1
00
014
DUP1
014
DUP1
014
DUP1
015
REVERT

015
REVERT
015
REVERT
016
JUMPDEST
016
JUMPDEST
016
JUMPDEST
017
POP
017
POP
017
POP
018
PUSH1
40
018
PUSH1
40
018
PUSH1
40
020
MLOAD
020
MLOAD
020
MLOAD
021
PUSH1
20
021
PUSH1
20
021
PUSH1
20
023
DUP1
023
DUP1
023
DUP1
024
PUSH2
0217
024
PUSH2
0217
024
PUSH2
0217
027
DUP4
027
DUP4
027

DUP4
028
CODECOPY
028
CODECOPY
028
CODECOPY
029
DUP2
029
DUP2
029
DUP2
030
ADD
030
ADD
030
ADD
031
PUSH1
40
031
PUSH1
40
031
PUSH1
40
033
SWAP1
033
SWAP1
033
SWAP1
034
DUP2
034
DUP2
034
DUP2
035
MSTORE
035
MSTORE
035
MSTORE
036
SWAP1
036
SWAP1
036
SWAP1
037
MLOAD
037
MLOAD
037
MLOAD
038
PUSH1

00

038

PUSH1

00

038

PUSH1

00

040

DUP2

040

DUP2

040

DUP2

041

DUP2

041

DUP2

041

DUP2

042

SSTORE

042

SSTORE

042

SSTORE

043

CALLER

043

CALLER

043

CALLER

044

DUP2

044

DUP2

044

DUP2

045

MSTORE

045

MSTORE

045

MSTORE

046

PUSH1

01

046

PUSH1

01

046

PUSH1

01

048

PUSH1

20

048

PUSH1

20

048

PUSH1

20
050
MSTORE
050
MSTORE
050
MSTORE
051
SWAP2
051
SWAP2
051
SWAP2
052
SWAP1
052
SWAP1
052
SWAP1
053
SWAP2
053
SWAP2
053
SWAP2
054
SHA3
054
SHA3
054
SHA3
055
SSTORE
055
SSTORE
055
SSTORE
056
PUSH2 01d1
056
PUSH2 01d1
056
PUSH2 01d1
059
DUP1
059
DUP1
059
DUP1
060
PUSH2
0046
060
PUSH2
0046
060
PUSH2
0046
063
PUSH1

00
063
PUSH1
00
063
PUSH1
00
065
CODECOPY
065
CODECOPY
065
CODECOPY
066
PUSH1
00
066
PUSH1
00
066
PUSH1
00
068
RETURN
068
RETURN
068
RETURN
069
STOP
069
STOP
069
STOP
070
PUSH1
80
070
PUSH1
80
070
PUSH1
80
072
PUSH1
40
072
PUSH1
40
072
PUSH1
40
074
MSTORE
074
MSTORE
074
MSTORE
075
PUSH1

04

075

PUSH1

04

075

PUSH1

04

077

CALLDATASIZE

077

CALLDATASIZE

077

CALLDATASIZE

078

LT

078

LT

078

LT

079

PUSH2

0056

079

PUSH2

0056

079

PUSH2

0056

082

JUMPI

082

JUMPI

082

JUMPI

083

PUSH4 ffffffff

083

PUSH4 ffffffff

083

PUSH4 ffffffff

088

PUSH29

01000000000000000000000000000000000000000000000000000000000000

088

PUSH29

01000000000000000000000000000000000000000000000000000000000000

088

PUSH29

01000000000000000000000000000000000000000000000000000000000000

118

PUSH1

00

118

PUSH1

00

118

PUSH1

00

120

CALLDATALOAD

120
CALLDATALOAD
120
CALLDATALOAD
121
DIV
121
DIV
121
DIV
122
AND
122
AND
122
AND
123
PUSH4 18160ddd
123
PUSH4 18160ddd
123
PUSH4 18160ddd
128
DUP2
128
DUP2
128
DUP2
129
EQ
129
EQ
129
EQ
130
PUSH2 005b
130
PUSH2 005b
130
PUSH2 005b
133
JUMPI
133
JUMPI
133
JUMPI
134
DUP1
134
DUP1
134
DUP1
135
PUSH4 70a08231
135
PUSH4 70a08231
135
PUSH4 70a08231
140
EQ

140
EQ
140
EQ
141
PUSH2
0082
141
PUSH2
0082
141
PUSH2
0082
144
JUMPI
144
JUMPI
144
JUMPI
145
DUP1
145
DUP1
145
DUP1
146
PUSH4 a9059cbb
146
PUSH4 a9059cbb
146
PUSH4 a9059cbb
151
EQ
151
EQ
151
EQ
152
PUSH2 00b0
152
PUSH2 00b0
152
PUSH2 00b0
155
JUMPI
155
JUMPI
155
JUMPI
156
JUMPDEST
156
JUMPDEST
156
JUMPDEST
157
PUSH1
00
157
PUSH1

00

157

PUSH1

00

159

DUP1

159

DUP1

159

DUP1

160

REVERT

160

REVERT

160

REVERT

161

JUMPDEST

161

JUMPDEST

161

JUMPDEST

162

CALLVALUE

162

CALLVALUE

162

CALLVALUE

163

DUP1

163

DUP1

163

DUP1

164

ISZERO

164

ISZERO

164

ISZERO

165

PUSH2

0067

165

PUSH2

0067

165

PUSH2

0067

168

JUMPI

168

JUMPI

168

JUMPI

169

PUSH1

00

169

PUSH1

00

169

PUSH1

00

171

DUP1

171

DUP1

171

DUP1

172

REVERT

172

REVERT

172

REVERT

173

JUMPDEST

173

JUMPDEST

173

JUMPDEST

174

POP

174

POP

174

POP

175

PUSH2

0070

175

PUSH2

0070

175

PUSH2

0070

178

PUSH2 00f5

178

PUSH2 00f5

178

PUSH2 00f5

181

JUMP

181

JUMP

181

JUMP

182

JUMPDEST

182

JUMPDEST

182

JUMPDEST

183

PUSH1

40

183

PUSH1

40
183
PUSH1
40
185
DUP1
185
DUP1
185
DUP1
186
MLOAD
186
MLOAD
186
MLOAD
187
SWAP2
187
SWAP2
187
SWAP2
188
DUP3
188
DUP3
188
DUP3
189
MSTORE
189
MSTORE
189
MSTORE
190
MLOAD
190
MLOAD
190
MLOAD
191
SWAP1
191
SWAP1
191
SWAP1
192
DUP2
192
DUP2
192
DUP2
193
SWAP1
193
SWAP1
193
SWAP1
194
SUB

194
SUB
194
SUB
195
PUSH1
20
195
PUSH1
20
195
PUSH1
20
197
ADD
197
ADD
197
ADD
198
SWAP1
198
SWAP1
198
SWAP1
199
RETURN
199
RETURN
199
RETURN
200
JUMPDEST
200
JUMPDEST
200
JUMPDEST
201
CALLVALUE
201
CALLVALUE
201
CALLVALUE
202
DUP1
202
DUP1
202
DUP1
203
ISZERO
203
ISZERO
203
ISZERO
204
PUSH2 008e
204
PUSH2 008e
204

PUSH2 008e

207

JUMPI

207

JUMPI

207

JUMPI

208

PUSH1

00

208

PUSH1

00

208

PUSH1

00

210

DUP1

210

DUP1

210

DUP1

211

REVERT

211

REVERT

211

REVERT

212

JUMPDEST

212

JUMPDEST

212

JUMPDEST

213

POP

213

POP

213

POP

214

PUSH2

0070

214

PUSH2

0070

214

PUSH2

0070

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

217

PUSH20 ffffffffffffffffffffffffffffffffffffffff

238

PUSH1

04

238

PUSH1

04
238
PUSH1
04
240
CALLDATALOAD
240
CALLDATALOAD
240
CALLDATALOAD
241
AND
241
AND
241
AND
242
PUSH2 00fb
242
PUSH2 00fb
242
PUSH2 00fb
245
JUMP
245
JUMP
245
JUMP
246
JUMPDEST
246
JUMPDEST
246
JUMPDEST
247
CALLVALUE
247
CALLVALUE
247
CALLVALUE
248
DUP1
248
DUP1
248
DUP1
249
ISZERO
249
ISZERO
249
ISZERO
250
PUSH2 00bc
250
PUSH2 00bc
250
PUSH2 00bc
253
JUMPI

253
JUMPI
253
JUMPI
254
PUSH1
00
254
PUSH1
00
254
PUSH1
00
256
DUP1
256
DUP1
256
DUP1
257
REVERT
257
REVERT
257
REVERT
258
JUMPDEST
258
JUMPDEST
258
JUMPDEST
259
POP
259
POP
259
POP
260
PUSH2 00e1
260
PUSH2 00e1
260
PUSH2 00e1
263
PUSH20 ffffffffffffffffffffffffffffffffffffffff
263
PUSH20 ffffffffffffffffffffffffffffffffffffffff
263
PUSH20 ffffffffffffffffffffffffffffffffffffffff
284
PUSH1
04
284
PUSH1
04
284
PUSH1
04
286
CALLDATALOAD

286
CALLDATALOAD
286
CALLDATALOAD
287
AND
287
AND
287
AND
288
PUSH1
24
288
PUSH1
24
288
PUSH1
24
290
CALLDATALOAD
290
CALLDATALOAD
290
CALLDATALOAD
291
PUSH2
0123
291
PUSH2
0123
291
PUSH2
0123
294
JUMP
294
JUMP
294
JUMP
295
JUMPDEST
295
JUMPDEST
295
JUMPDEST
296
PUSH1
40
296
PUSH1
40
296
PUSH1
40
298
DUP1
298
DUP1
298

DUP1
299
MLOAD
299
MLOAD
299
MLOAD
300
SWAP2
300
SWAP2
300
SWAP2
301
ISZERO
301
ISZERO
301
ISZERO
302
ISZERO
302
ISZERO
302
ISZERO
303
DUP3
303
DUP3
303
DUP3
304
MSTORE
304
MSTORE
304
MSTORE
305
MLOAD
305
MLOAD
305
MLOAD
306
SWAP1
306
SWAP1
306
SWAP1
307
DUP2
307
DUP2
307
DUP2
308
SWAP1
308
SWAP1
308

SWAP1

309

SUB

309

SUB

309

SUB

310

PUSH1

20

310

PUSH1

20

310

PUSH1

20

312

ADD

312

ADD

312

ADD

313

SWAP1

313

SWAP1

313

SWAP1

314

RETURN

314

RETURN

314

RETURN

315

JUMPDEST

315

JUMPDEST

315

JUMPDEST

316

PUSH1

00

316

PUSH1

00

316

PUSH1

00

318

SLOAD

318

SLOAD

318

SLOAD

319

SWAP1

319

SWAP1

319

SWAP1

320

JUMP

320

JUMP

320

JUMP

321

JUMPDEST

321

JUMPDEST

321

JUMPDEST

322

PUSH20 ffffffffffffffffffffffffffffffffffffffff

322

PUSH20 ffffffffffffffffffffffffffffffffffffffff

322

PUSH20 ffffffffffffffffffffffffffffffffffffffff

343

AND

343

AND

343

AND

344

PUSH1

00

344

PUSH1

00

344

PUSH1

00

346

SWAP1

346

SWAP1

346

SWAP1

347

DUP2

347

DUP2

347

DUP2

348

MSTORE

348

MSTORE

348

MSTORE

349

PUSH1

01

349

PUSH1

01

349

PUSH1

01

351

PUSH1

20

351

PUSH1

20

351

PUSH1

20

353

MSTORE

353

MSTORE

353

MSTORE

354

PUSH1

40

354

PUSH1

40

354

PUSH1

40

356

SWAP1

356

SWAP1

356

SWAP1

357

SHA3

357

SHA3

357

SHA3

358

SLOAD

358

SLOAD

358

SLOAD

359

SWAP1

359

SWAP1

359

SWAP1

360

JUMP

360

JUMP

360

JUMP

361

JUMPDEST

361

JUMPDEST

361

JUMPDEST

362

PUSH1

00

362

PUSH1

00

362

PUSH1

00

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

364

PUSH20 ffffffffffffffffffffffffffffffffffffffff

385

DUP4

385

DUP4

385

DUP4

386

AND

386

AND

386

AND

387

ISZERO

387

ISZERO

387

ISZERO

388

ISZERO

388

ISZERO

388

ISZERO

389

PUSH2

0147

389

PUSH2

0147

389

PUSH2

0147

392

JUMPI

392

JUMPI

392

JUMPI

393

PUSH1

00

393

PUSH1

00
393
PUSH1
00
395
DUP1
395
DUP1
395
DUP1
396
REVERT
396
REVERT
396
REVERT
397
JUMPDEST
397
JUMPDEST
397
JUMPDEST
398
CALLER
398
CALLER
398
CALLER
399
PUSH1
00
399
PUSH1
00
399
PUSH1
00
401
SWAP1
401
SWAP1
401
SWAP1
402
DUP2
402
DUP2
402
DUP2
403
MSTORE
403
MSTORE
403
MSTORE
404
PUSH1
01
404
PUSH1

01

404

PUSH1

01

406

PUSH1

20

406

PUSH1

20

406

PUSH1

20

408

MSTORE

408

MSTORE

408

MSTORE

409

PUSH1

40

409

PUSH1

40

409

PUSH1

40

411

SWAP1

411

SWAP1

411

SWAP1

412

SHA3

412

SHA3

412

SHA3

413

SLOAD

413

SLOAD

413

SLOAD

414

DUP3

414

DUP3

414

DUP3

415

GT

415

GT

415

GT

416

ISZERO

416
ISZERO
416
ISZERO
417
PUSH2
0163
417
PUSH2
0163
417
PUSH2
0163
420
JUMPI
420
JUMPI
420
JUMPI
421
PUSH1
00
421
PUSH1
00
421
PUSH1
00
423
DUP1
423
DUP1
423
DUP1
424
REVERT
424
REVERT
424
REVERT
425
JUMPDEST
425
JUMPDEST
425
JUMPDEST
426
POP
426
POP
426
POP
427
CALLER
427
CALLER
427
CALLER
428
PUSH1

00
428
PUSH1
00
428
PUSH1
00
430
SWAP1
430
SWAP1
430
SWAP1
431
DUP2
431
DUP2
431
DUP2
432
MSTORE
432
MSTORE
432
MSTORE
433
PUSH1
01
433
PUSH1
01
433
PUSH1
01
435
PUSH1
20
435
PUSH1
20
435
PUSH1
20
437
DUP2
437
DUP2
437
DUP2
438
SWAP1
438
SWAP1
438
SWAP1
439
MSTORE
439
MSTORE
439

MSTORE

440

PUSH1

40

440

PUSH1

40

440

PUSH1

40

442

DUP1

442

DUP1

442

DUP1

443

DUP4

443

DUP4

443

DUP4

444

SHA3

444

SHA3

444

SHA3

445

DUP1

445

DUP1

445

DUP1

446

SLOAD

446

SLOAD

446

SLOAD

447

DUP6

447

DUP6

447

DUP6

448

SWAP1

448

SWAP1

448

SWAP1

449

SUB

449

SUB

449

SUB

450

SWAP1

450

SWAP1

450

SWAP1

451

SSTORE

451

SSTORE

451

SSTORE

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

452

PUSH20 ffffffffffffffffffffffffffffffffffffffff

473

DUP6

473

DUP6

473

DUP6

474

AND

474

AND

474

AND

475

DUP4

475

DUP4

475

DUP4

476

MSTORE

476

MSTORE

476

MSTORE

477

SWAP1

477

SWAP1

477

SWAP1

478

SWAP2

478

SWAP2

478

SWAP2

479

SHA3

479

SHA3

479

SHA3

480

DUP1

480
DUP1
480
DUP1
481
SLOAD
481
SLOAD
481
SLOAD
482
DUP4
482
DUP4
482
DUP4
483
ADD
483
ADD
483
ADD
484
SWAP1
484
SWAP1
484
SWAP1
485
SSTORE
485
SSTORE
485
SSTORE
486
SWAP3
486
SWAP3
486
SWAP3
487
SWAP2
487
SWAP2
487
SWAP2
488
POP
488
POP
488
POP
489
POP
489
POP
489
POP
490
JUMP

490
JUMP
490
JUMP
491
STOP
491
STOP
491
STOP
492
LOG1
492
LOG1
492
LOG1
493
PUSH6 627a7a723058
493
PUSH6 627a7a723058
493
PUSH6 627a7a723058
500
SHA3
500
SHA3
500
SHA3
501
INVALID
501
INVALID
501
INVALID
502
INVALID
502
INVALID
502
INVALID
503
SWAP10
503
SWAP10
503
SWAP10
504
DELEGATECALL
504
DELEGATECALL
504
DELEGATECALL
505
GASLIMIT
505
GASLIMIT
505
GASLIMIT
506
SWAP7

506
SWAP7
506
SWAP7
507
TIMESTAMP
507
TIMESTAMP
507
TIMESTAMP
508
DUP8
508
DUP8
508
DUP8
509
INVALID
509
INVALID
509
INVALID
510
INVALID
510
INVALID
510
INVALID
511
INVALID
511
INVALID
511
INVALID
512
SWAP4
512
SWAP4
512
SWAP4
513
LOG4
513
LOG4
513
LOG4
514
SWAP1
514
SWAP1
514
SWAP1
515
JUMPI
515
JUMPI
515
JUMPI
516
INVALID

516
INVALID
516
INVALID
517
CALLVALUE
517
CALLVALUE
517
CALLVALUE
518
INVALID
518
INVALID
518
INVALID
519
BLOCKHASH
519
BLOCKHASH
519
BLOCKHASH
520
INVALID
520
INVALID
520
INVALID
521
SWAP2
521
SWAP2
521
SWAP2
522
INVALID
522
INVALID
522
INVALID
523
INVALID
523
INVALID
523
INVALID
524
INVALID
524
INVALID
524
INVALID
525
INVALID
525
INVALID
525
INVALID
526
CALLCODE

526
CALLCODE
526
CALLCODE
527
SWAP13
527
SWAP13
527
SWAP13
528
DUP9
528
DUP9
528
DUP9
529
INVALID
529
INVALID
529
INVALID
530
INVALID
530
INVALID
530
INVALID
531
RETURN
531
RETURN
531
RETURN
532
INVALID
532
INVALID
532
INVALID
533
STOP
533
STOP
533
STOP
534
INVALID
534
INVALID
534
INVALID
535
STOP
535
STOP
535
STOP
536
STOP

536
STOP
536
STOP
537
STOP
537
STOP
537
STOP
538
STOP
538
STOP
538
STOP
539
STOP
539
STOP
539
STOP
540
STOP
540
STOP
540
STOP
541
STOP
541
STOP
541
STOP
542
STOP
542
STOP
542
STOP
543
STOP
543
STOP
543
STOP
544
STOP
544
STOP
544
STOP
545
STOP
545
STOP
545
STOP
546
STOP

546
STOP
546
STOP
547
STOP
547
STOP
547
STOP
548
STOP
548
STOP
548
STOP
549
STOP
549
STOP
549
STOP
550
STOP
550
STOP
550
STOP
551
STOP
551
STOP
551
STOP
552
STOP
552
STOP
552
STOP
553
STOP
553
STOP
553
STOP
554
STOP
554
STOP
554
STOP
555
STOP
555
STOP
555
STOP
556
STOP

556
STOP
556
STOP
557
STOP
557
STOP
557
STOP
558
STOP
558
STOP
558
STOP
559
STOP
559
STOP
559
STOP
560
STOP
560
STOP
560
STOP
561
STOP
561
STOP
561
STOP
562
STOP
562
STOP
562
STOP
563
STOP
563
STOP
563
STOP
564
STOP
564
STOP
564
STOP
565
INVALID
565
INVALID
565
INVALID
566
LT

566

LT

566

LT

view raw

[

BasicToken.asm

](https://gist.github.com/ajsantander/60bd8d6f88725663f89a67a7c51672c7#file-basictoken-asm)

hosted with ❤ by GitHub

view raw

[

BasicToken.asm

](https://gist.github.com/ajsantander/60bd8d6f88725663f89a67a7c51672c7#file-basictoken-asm)

GitHubTo make sure that you're following the same set of opcodes described in this series, please compare what you see in Remix with the bytecode in this gist.

bytecode in this gistThis is the disassembled bytecode of the contract. Disassembly sounds rather intimidating, but it's quite simple, really. If you scan the raw bytecode by bytes (two characters at a time), the EVM identifies specific opcodes that it associates to particular actions. For example:

0x60 => PUSH 0x01 => ADD 0x02 => MUL 0x00 => STOP ...

The disassembled code is still very low-level and difficult to read, but as you will see, we can start making sense out of it.

Opcodes

Before we get started on our ambitious endeavour of completely deconstructing the bytecode, you're going to need a basic tool set for understanding individual opcodes such as PUSH

, ADD

, SWAP

, DUP

, etc. An opcode, in the end, can only push or consume items from the EVM's stack, memory, or storage belonging to the contract. That's it.

PUSH

ADD

SWAP

DUP

To see all the available opcodes that the EVM can process, check outthis handy gist from Pyethereum showing a list of the opcodes To understand what each one does and how it works,Solidity's assembly documentation is a great reference. Even though it's not a one-on-one relationship with the raw opcodes, it's pretty close (it's actually Yul, an intermediate language between Solidity and EVM bytecode). Finally, if you can speak scientician, there's always the Ethereum Yellow Paper to fall back on.

this handy gist from Pyethereum showing a list of the opcodesSolidity's assembly documentationEthereum Yellow PaperThere's no point in reading these resources from start to finish right now; just keep them around for reference. We'll be using them as we go along.

Instructions

Each line in the disassembled code above is an instruction for the EVM to execute. Each instruction contains an opcode. For example, let's take one of those instructions, instruction 88, which pushes the number 4 to the stack. This particular disassembler interprets instructions as follows:

88 PUSH1 0x04 | | |
| | Hex value for push. | Opcode. Instruction number.

Even though the disassembled code brings us one step closer to understanding what's going on, it's still quite intimidating. We're going to need a strategy for deconstructing the whole thing, which has 596 instructions!

The Strategy

Problems that appear to be overwhelming at first usually succumb to the all-powerful, all-mighty "divide-and-conquer" strategy, and this problem is no exception to the rule. We'll identify split points in the disassembled code and reduce it bit by bit, until we end up with small, digestible chunks, which we'll walk through step by step in Remix's debugger. In the following diagram, we can see the first split we can make on the disassembled code, which we'll analyze completely in the next article.

You can find the end result of the entire deconstruction in thedeconstruction diagram. Don't worry if you don't understand the diagram at first. You're not supposed to. This series will go through it step by step. Keep it around so you can keep track of the big picture as we go along.

deconstruction diagramThe series is divided into the following set of articles. If you're up for the challenge, get started with the actual deconstruction inPart II. See you there!

Part II1. Deconstructing a Solidity Contract—Part I: Introduction ✔

1. Deconstructing a Solidity Contract—Part II: Creation vs. Runtime
2. Deconstructing a Solidity Contract—Part III: The Function Selector
3. Deconstructing a Solidity Contract—Part IV: Function Wrappers
4. Deconstructing a Solidity Contract—Part V: Function Bodies
5. Deconstructing a Solidity Contract—Part VI: The Metadata Hash

Deconstructing a Solidity Contract—Part I: Introduction ✔

Deconstructing a Solidity Contract—Part II: Creation vs. Runtime

Deconstructing a Solidity Contract—Part II: Creation vs. RuntimeDeconstructing a Solidity Contract—Part III: The Function Selector

Deconstructing a Solidity Contract—Part III: The Function SelectorDeconstructing a Solidity Contract—Part IV: Function Wrappers

Deconstructing a Solidity Contract—Part IV: Function WrappersDeconstructing a Solidity Contract—Part V: Function Bodies

Deconstructing a Solidity Contract—Part V: Function BodiesDeconstructing a Solidity Contract—Part VI: The Metadata Hash

Deconstructing a Solidity Contract—Part VI: The Metadata Hash