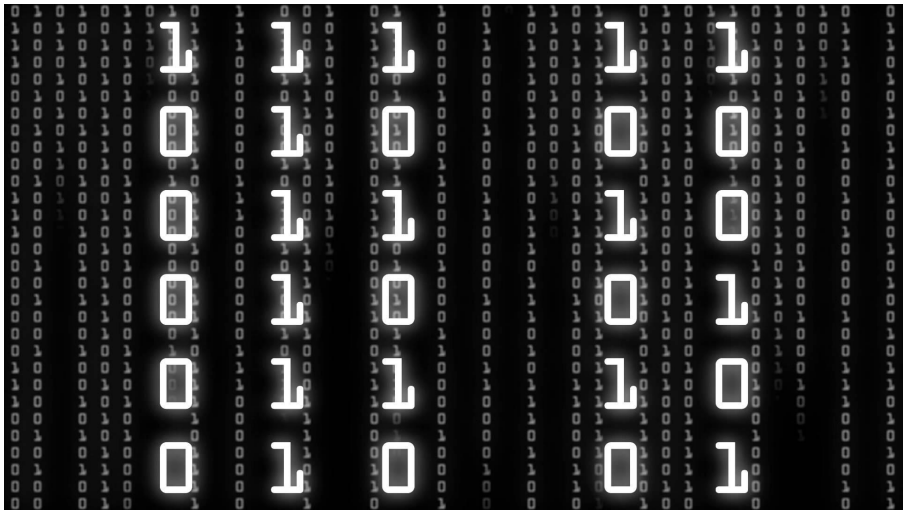
Number Systems

Binary Number System

- One of two numbers: **0** or **1**

- Represents capacitors being off (0) or on (1)



Binary Number System

- Represents the base of computer knowledge
  - Computers, on the most basic level, can only read 1’s and 0’s
  - So we have to translate between them so we can understand what is occurring when we code

Binary Number System

- Translating between binary and base 10 (normal numbers)

- 1 0 1 0 = ??

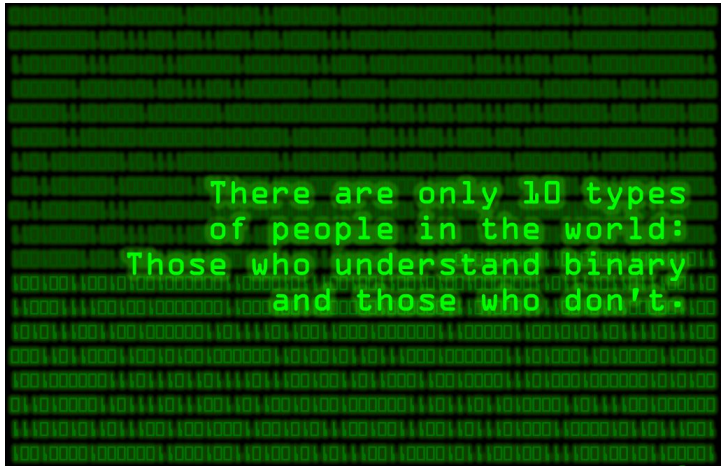
Binary Number System

- Translating between binary and base 10 (normal numbers)

- 1 0 1 0 (binary) = **10_{10} (base 10)**

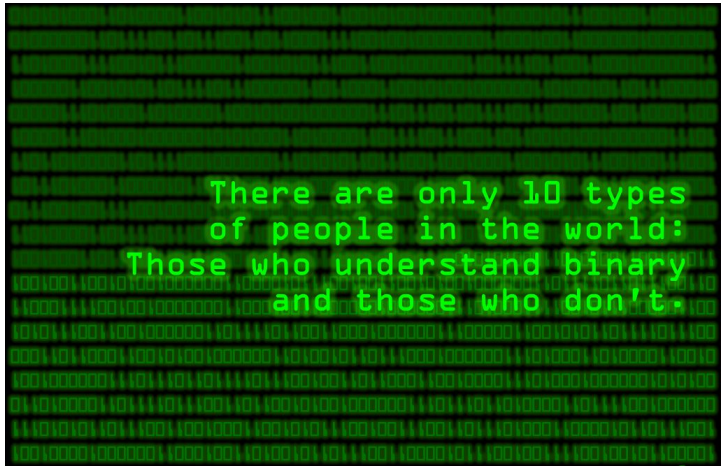
Binary Number System

- 1 1 1 0 0 1 0 1 = ?
- 1 0 0 0 0 = ?
- 1 1 1 1 1 1 1 1 1 = ?
- 1 0 0 0 0 1 1 = ?



Binary Number System

- $1\ 1\ 1\ 0\ 0\ 1\ 0\ 1 = 229_{10}$
- $1\ 0\ 0\ 0\ 0 = 16_{10}$
- $1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 = 511_{10}$
- $1\ 0\ 0\ 0\ 0\ 1\ 1 = 67_{10}$



Binary Number System

- Formalize it...write the algorithm of how to convert from binary to base 10

Binary Number System

- Converting from base 10 to binary

- 42_{10} = ??

Binary Number System

- Converting from base 10 to binary

- 42_{10} = 101010

Binary Number System

- 16_{10} = ?
- 513_{10} = ?
- 127_{10} = ?
- 99_{10} = ?

Binary Number System

- 16_{10} = 10000
- 513_{10} = 1000000001
- 127_{10} = 1111111
- 99_{10} = 1100011

Binary Number System

- Formalize it...write the algorithm of how to convert from base 10 to binary

Binary Number System

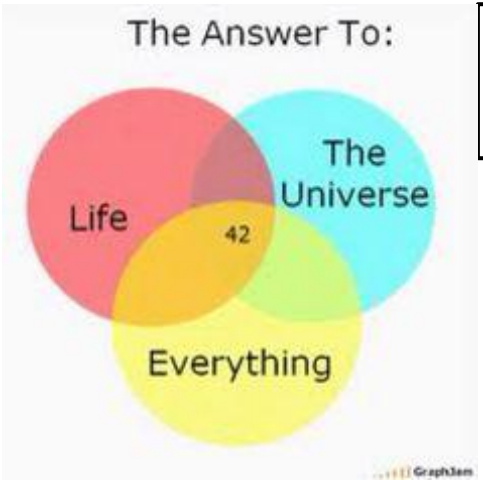
- Converting base 10 to binary - **Divide by Two Algorithm**

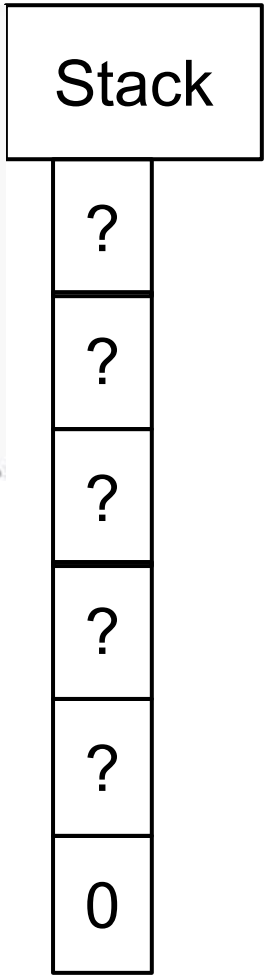
1) Assume the number is > 0
2) Divide the number by 2, write the remainder in a stack (bottom up)
3) When the number is reduced to zero, flip the stack. This is your binary number.

Divide by Two Algorithm

Start with 42

- 42 / 2 = 21, Remainder 0

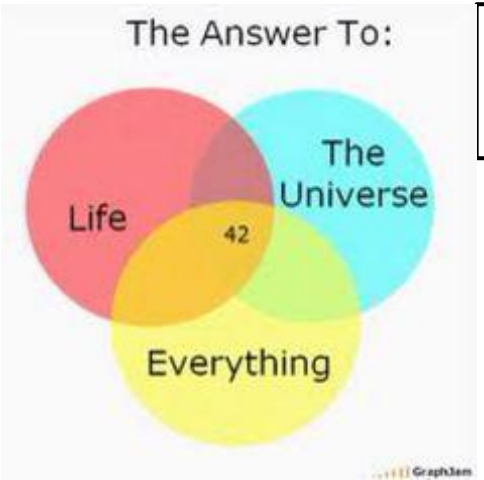


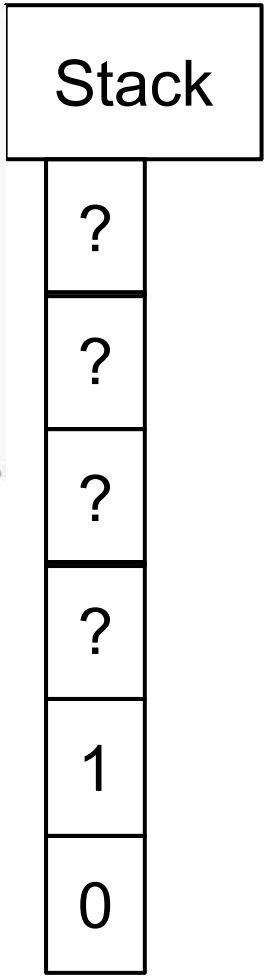


Divide by Two Algorithm

Start with 42

- 42 / 2 = 21, Remainder 0
- 21 / 2 = 10, Remainder 1

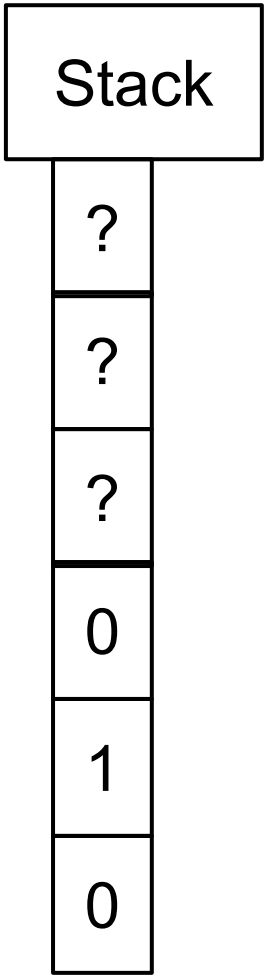




Divide by Two Algorithm

Start with 42

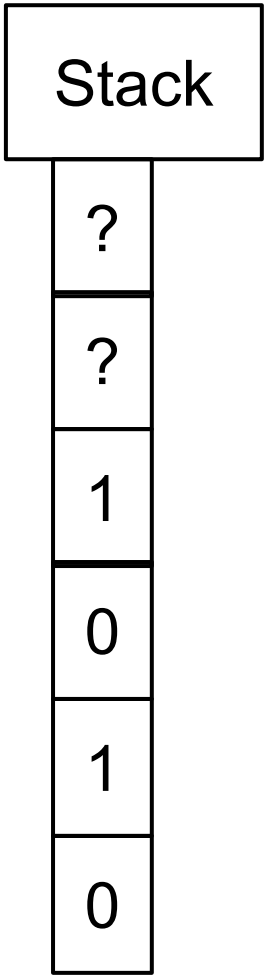
- 42 / 2 = 21, Remainder 0
- 21 / 2 = 10, Remainder 1
- 10 / 2 = 5 , Remainder 0



Divide by Two Algorithm

Start with 42

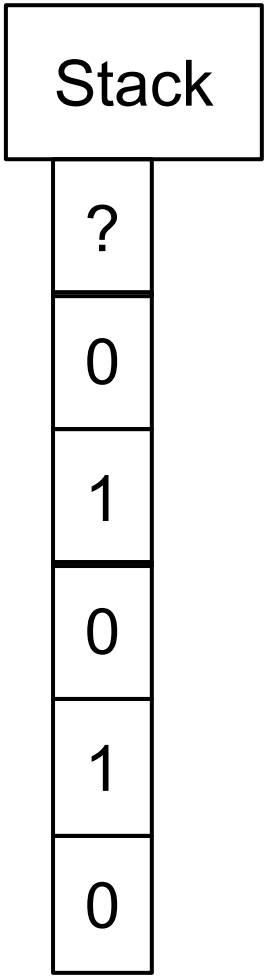
- 42 / 2 = 21, Remainder 0
- 21 / 2 = 10, Remainder 1
- 10 / 2 = 5 , Remainder 0
- 5 / 2 = 2 , Remainder 1



Divide by Two Algorithm

Start with 42

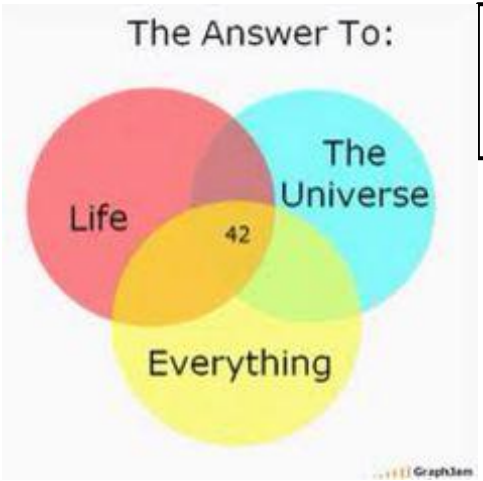
- 42 / 2 = 21, Remainder 0
- 21 / 2 = 10, Remainder 1
- 10 / 2 = 5 , Remainder 0
- 5 / 2 = 2 , Remainder 1
- 2 / 2 = 1 , Remainder 0

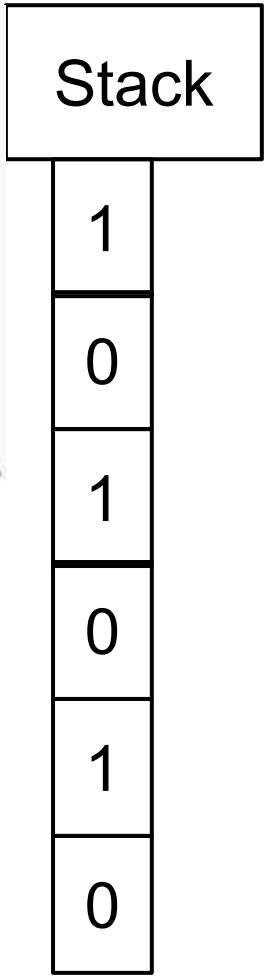


Divide by Two Algorithm

Start with 42

- 42 / 2 = 21, Remainder 0
- 21 / 2 = 10, Remainder 1
- 10 / 2 = 5 , Remainder 0
- 5 / 2 = 2 , Remainder 1
- 2 / 2 = 1 , Remainder 0
- 1 / 2 = 0 , Remainder 1





Hexadecimal

- Computers use binary at a base level…

  ...but it’s extremely difficult to read

- Example: Memory addresses

Hexadecimal

- Example: Memory addresses
- Must be used by programmers to ensure their code is being stored properly

- Possible address: 0011011111111100
- Another address: 0110011111011101
- Another address: 1100001111001111

Hexadecimal

- Using only binary to look through memory addresses is too much for a non-computer
- Solution = *hexadecimal*
  - **Base 16**
  - **Why base 16??**

Hexadecimal

- Why base 16?
  - Exponent of 2 ($2^4 = 16$)
  - Therefore, extremely easy to convert between values

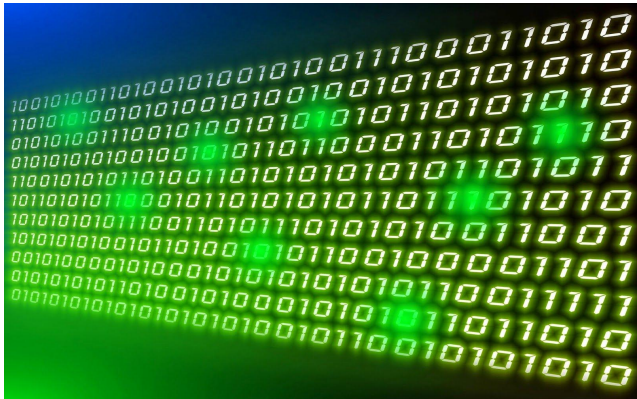
Hexadecimal Notation

Base 10	Hex	Base 10	Hex
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

Hexadecimal Notation

- Can write hex numbers in two ways:
  - $FF07_{16}$ *or* 0xFF07

  - First way → standard
  - Second way → C notation



Hexadecimal Conversions

- Hex to base 10
  - Same as binary
  - OxFF07: F F 0 7
    - 16^3 16^2 16^1 16^0

Hexadecimal Conversions

- Hex to base 10
  - Same as binary
  - OxFF07: F F 0 7

$$16^3 \quad 16^2 \quad 16^1 \quad 16^0$$

$$= \quad 4096 \quad 256 \quad 16 \quad 1$$

Hexadecimal Conversions

- Hex to base 10
  - Same as binary
  - OxFF07: F F 0 7
    - 16^3 16^2 16^1 16^0
    - = 15*4096 15*256 0*16 7*1

Hexadecimal Conversions

- Hex to base 10
  - Same as binary
  - OxFF07: F F 0 7
    - 16^3 16^2 16^1 16^0
    - = 15*4096 15*256 0*16 7*1
    - = 61,440 + 3840 + 0 + 7

Hexadecimal Conversions

- Hex to base 10
  - Same as binary
  - OxFF07: F F 0 7

$$16^3 \quad 16^2 \quad 16^1 \quad 16^0$$

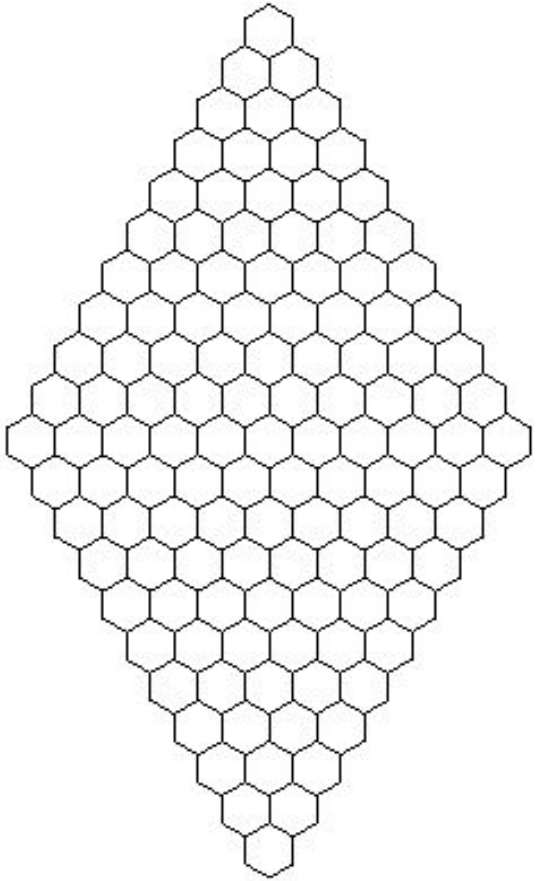
$$= 15*4096 \quad 15*256 \quad 0*16 \quad 7*1$$

$$= 61{,}440 + 3840 + 0 + 7$$

$$= \mathbf{65{,}287}$$

Hex to Base 10

- 0x56A = ?
- 0xBCD08 = ?
- 0x12AA = ?
- 0x10E = ?



Hex to Binary

- Each hex number corresponds to a 4-digit binary sequence

Binary	Hex	Decimal
	0	0
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
	A	10
	B	11
	C	12
	D	13
	E	14
	F	15

Hex to Binary

- To convert from hex to binary, replace the hex number with the binary sequence
- 0xED56 = ?
  - E = 1110
  - D = 1101
  - 5 = 0101
  - 6 = 0110

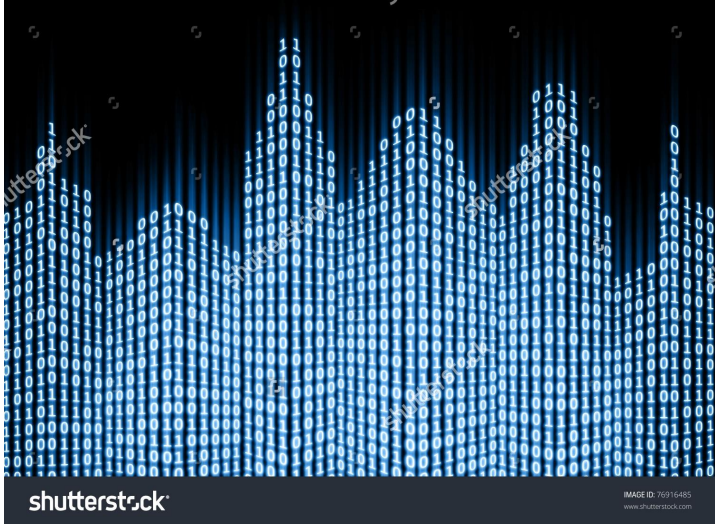
Hex to Binary

- To convert from hex to binary, replace the hex number with the binary sequence
- 0xED56 = ?
  - E = 1110
  - D = 1101
  - 5 = 0101
  - 6 = 0110

Therefore, 0xED56 = 1110110101010110

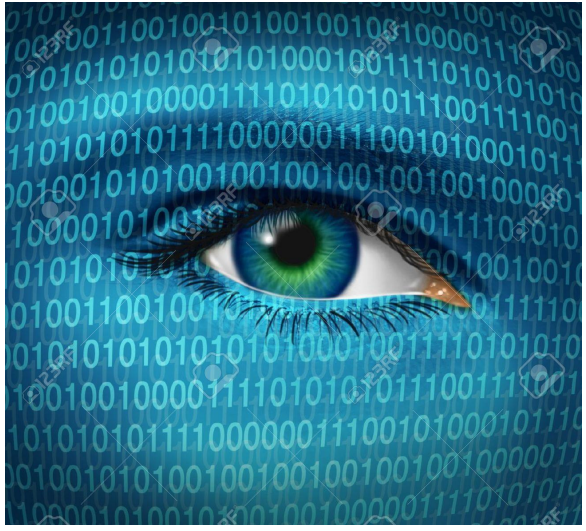
Hex to Binary

- 0xEEE5 = ?
- 0x34 = ?
- 0xCF078 = ?



Binary to Hex

- `0011 = ?`
- `0011110011111101 = ?`
- `1010110000100110010 1 = ?`



Decimal to Hex

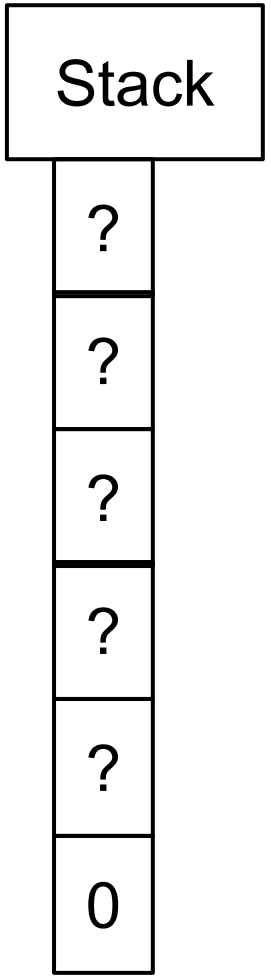
- Converting base 10 to hex- **Divide by 16 Algorithm**

1) Assume the number is > 0
2) Divide the number by 16, write the remainder in a stack (bottom up)- convert to hex notation if necessary
3) When the number is reduced to zero, flip the stack. This is your hex number.

Divide by 16 Algorithm

Start with 42000

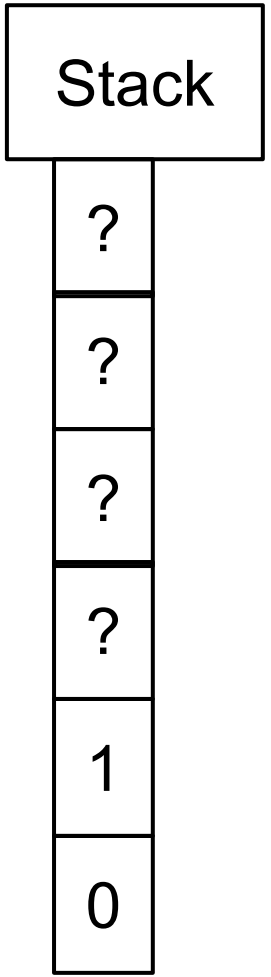
- 42 / 16 = 2625, Remainder 0



Divide by 16 Algorithm

Start with 42000

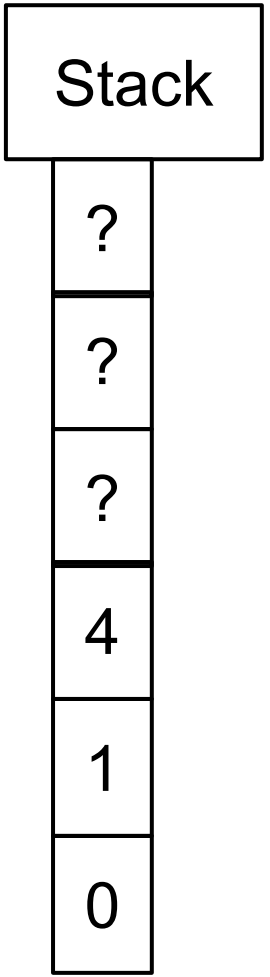
- 42000 / 16 = 2625, Remainder 0
- 2625 / 16 = 164, Remainder 1



Divide by 16 Algorithm

Start with 42000

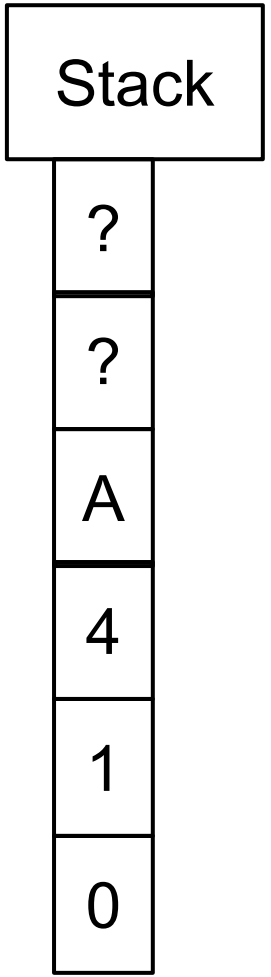
- 42000 / 16 = 2625, Remainder 0
- 2625 / 16 = 164, Remainder 1
- 164 / 16 = 10, Remainder 4



Divide by 16 Algorithm

Start with 42000

- 42000 / 16 = 2625, Remainder 0
- 2625 / 16 = 164, Remainder 1
- 164 / 16 = 10, Remainder 4
- 10 / 16 = 0, Remainder 10



Divide by 16 Algorithm

Start with 42000

- 42000 / 16 = 2625, Remainder 0
- 2625 / 16 = 164, Remainder 1
- 164 / 16 = 10, Remainder 4
- 10 / 16 = 0, Remainder 10

42000 = A410