

# Pointers in C

Md Tanvir Arafin

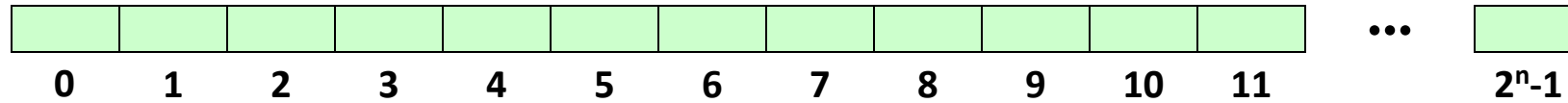
# Goals

- Demystify pointer operation in C

# Background

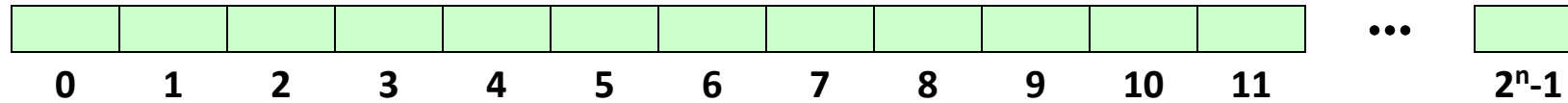
- Write basic C program
- Declare a variable
- Simple arithmetic

# Memory Organization (Simplified)



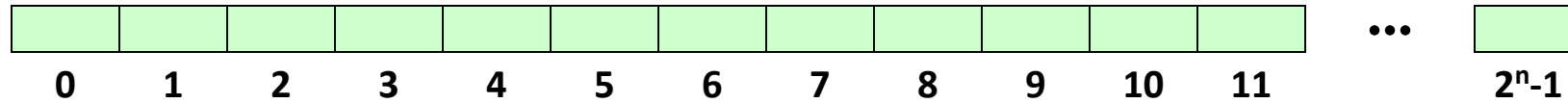
- Addressable memory cells
  - *Byte*: an 8-bit memory cell
  - *Address*: number by which a memory cell is identified

# Memory Organization (Simplified)



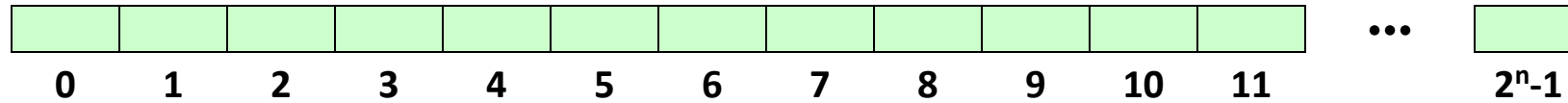
- Data types
  - **char** 1 byte
  - **short int** 2 bytes
  - **int** 4 bytes
  - **float** 4 bytes

# Memory Organization (Simplified)



- Data types
  - **char** 1 byte
  - **short int** 2 bytes
  - **int** 4 bytes
  - **float** 4 bytes

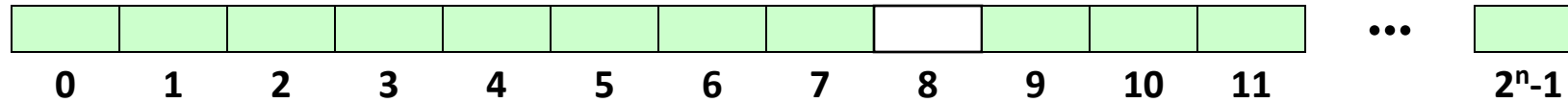
# Memory Organization (Simplified)



- What happens when we declare:

**char c;**

# Memory Organization (Simplified)

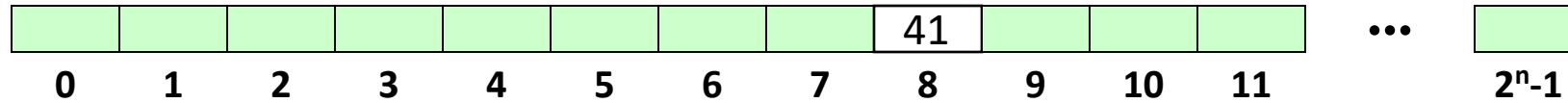


- What happens when we declare:

**char c;**



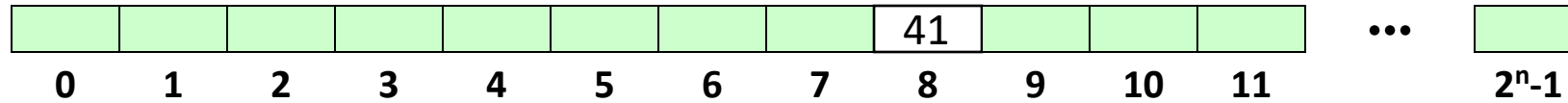
# Memory Organization (Simplified)



- What happens when we declare:

```
char c;  
c = 'A';
```

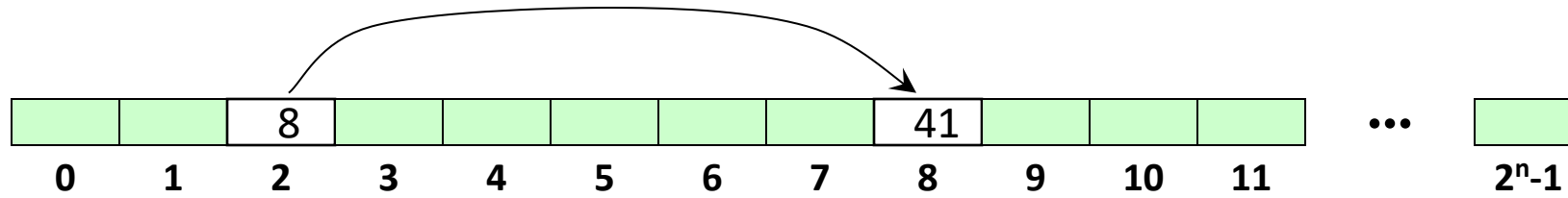
# Pointers



Can we know the address of the variable? Yes!!!

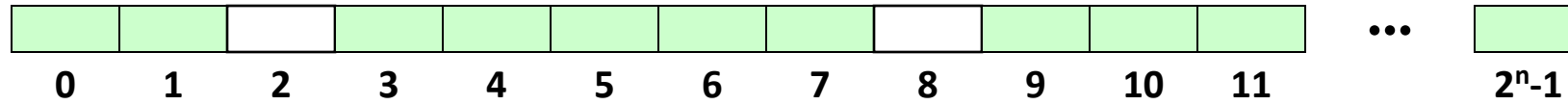
Pointer : A variable that contains the address of other variables

# Pointers



```
p = &c; /*assign address of c to variable p */
```

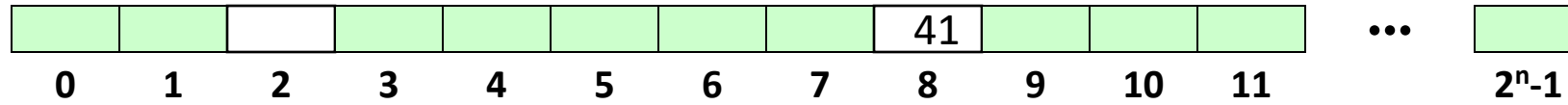
# Syntax



```
char c;
```

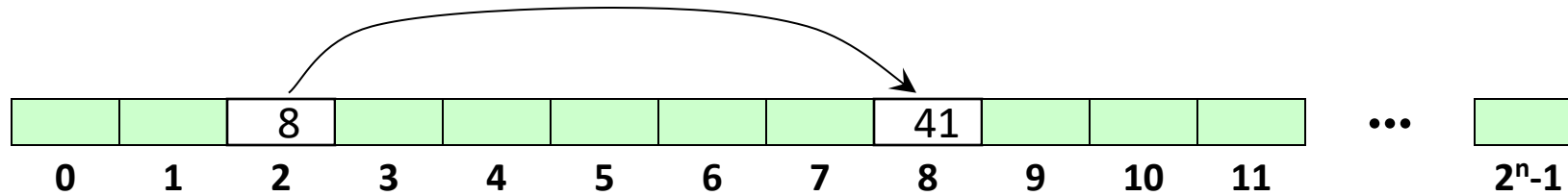
```
char* p; /* p is a pointer to char */
```

# Syntax



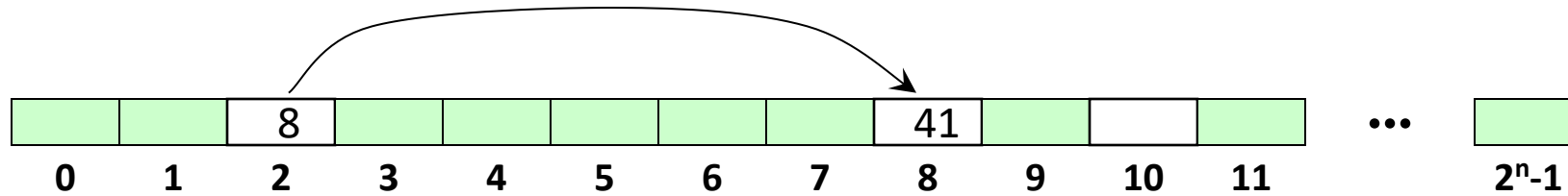
```
char c;  
char* p; /* p is a pointer to char */  
c = 'A';
```

# Syntax



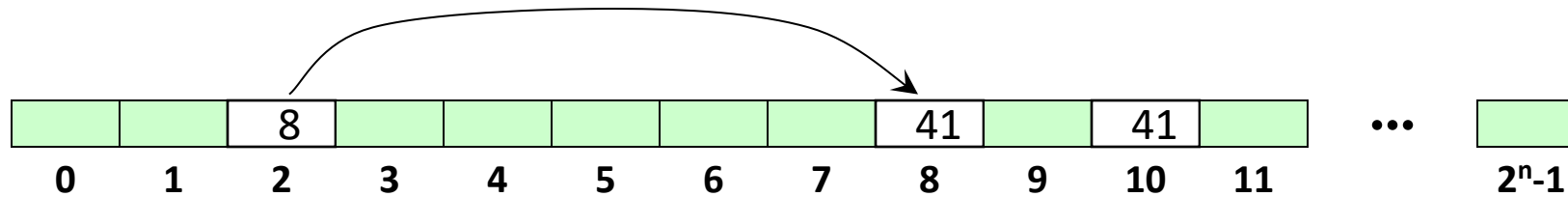
```
char c;  
char* p; /* p is a pointer to char */  
c = 'A';  
p = &c; /* p now points to c */
```

# Syntax



```
char c;  
char* p; /* p is a pointer to char */  
c = 'A';  
p = &c; /* p now points to c */  
char x;
```

# Syntax



```
char c;  
char* p; /* p is a pointer to char */  
c = 'A';  
p = &c; /* p now points to c */  
char x;  
x = *p; /* x is now 'A' */
```

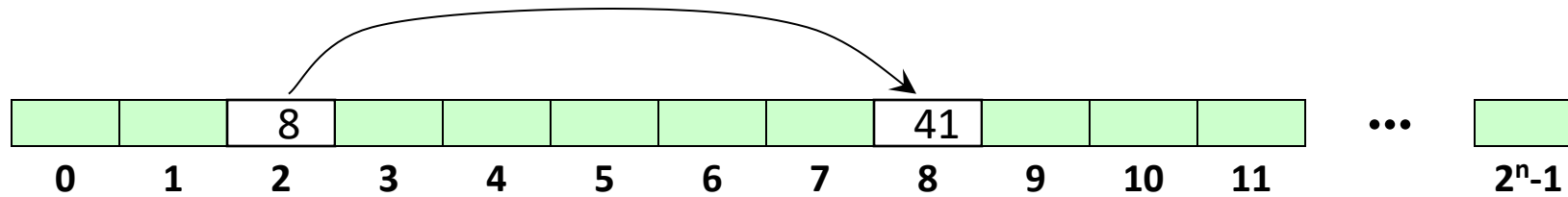
**\*** => Indirection/ dereferencing operator



# Syntax

- Unary '\*' operator *dereferences* a pointer
- Unary '&' operator generates a *pointer*
  - Operand of '&' must be an *l-value*

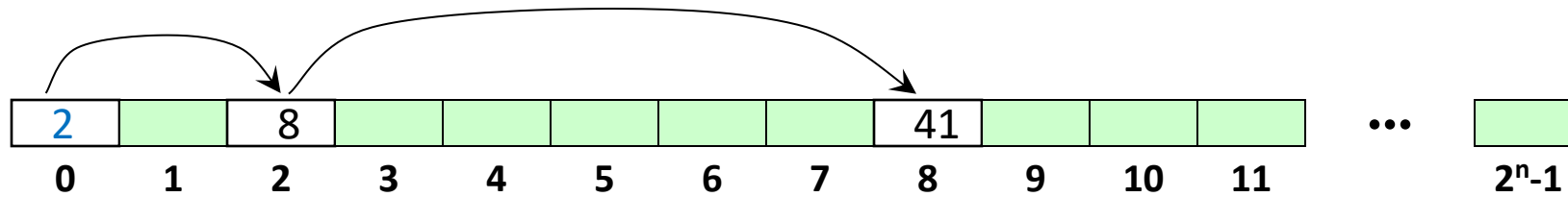
# Syntax: Quiz



```
char c;  
char* p;  
c = 'A';  
p = &c;
```

What would be the output if we print **c** and **p**?

# Syntax: Quiz



```
char c;  
char* p;  
char **q;  
c = 'A';  
p = &c;  
q = &p;
```

What would be the output if we print **q**?

# Demo