

# Function Input



# Inputs

```
void print_char(char c);  
int main(void);  
    char letter = "A";  
    print_char(letter);  
  
    return 0;  
}
```

```
void print_char(char c);  
    printf("%c", c);  
}
```

- Inputs are *passed into* a function
  - "char c" is the input to the function "print\_char"

# Inputs

```
void print_char(char c);  
int main(void);  
    char letter = "A";  
    print_char(letter);  
  
    return 0;  
}
```

```
void print_char(char c);  
    printf("%c", c);  
}
```

- REMINDER: Every variable in C has its own memory address!!
- Two variables in this code:
  - 1) Letter (type = char)
  - 2) C (type = char)

# Inputs

```
void print_char(char c);  
int main(void);  
    char letter = "A";  
    print_char(letter);  
  
    return 0;  
}
```

```
void print_char(char c);  
    printf("%c", c);  
}
```

- Two variables in this code:
  - 1) letter (type = char)
    - Is defined in main()
    - Can only be used in main()
  - 2) c (type = char)
    - Is defined in print\_char()
    - Can only be used in print\_char()

# Inputs

```
void print_char(char c);  
int main(void);  
    char letter = "A";  
    print_char(letter);  
  
    return 0;  
}
```

```
void print_char(char c);  
    printf("%c", c);  
}
```

- The value from **letter** is **copied over** to **c** when the `print_char()` function is called
  - Remain separate variables in memory
  - *Just happen to have the same value*

# Multiple Inputs

```
void add_nums(int num1, int num2,  
int num3);
```

```
int main(void) {  
    int n1 = 10;  
    int n2 = 20;  
    int n3 = 30;  
    add_nums(n1, n2, n3);  
    return 0;  
}
```

```
void add_nums(int num1, int num2,  
int num3) {  
    printf("The sum of the  
numbers is: %d\n", num1 + num2 +  
num3);  
}
```

- A function can have multiple inputs
  - There is theoretically no limit to the number of inputs to a function
  - Each variable is still a new memory location

# Input Coding Challenge 1



Scan in a char in `main()`, pass it to a function

Inside the function, if the char is a capital letter, print it out.

Otherwise, print “not a capital”

Continue scanning in chars in `main()` until the user enters a ‘!’

# Input Coding Challenge 2



Scan in three numbers in main()

Pass all three into two functions

1st function: Calculate the  
maximum value

2nd function: Calculate the  
minimum value