By: Emy Satira, Ilya, Naagajoothi

# C for Cats: Whiskers, Paws, and Programming

## SERIES 1: C FUNDAMENTAL



JABATAN KEJURUTERAAN ELEKTRIK
POLITEKNIK SULTAN SALAHUDDIN ABDUL AZIZ SHAH

**SERIES 1: C FUNDAMENTAL**

**WRITERS**

Emy Satira Azrin binti Mohamed Hakke

Ilya binti Ismail

Naagajoothi A/P Adin Naraina

**EDITORS**

Emy Satira Azrin binti Mohamed Hakke

Ilya binti Ismail

Naagajoothi A/P Adin Naraina

**First Publication 2024**

eISBN No: 978-629-7667-29-4

**Published by:**

**UNIT PENERBITAN**

Politeknik Sultan Salahuddin Abdul Aziz Shah

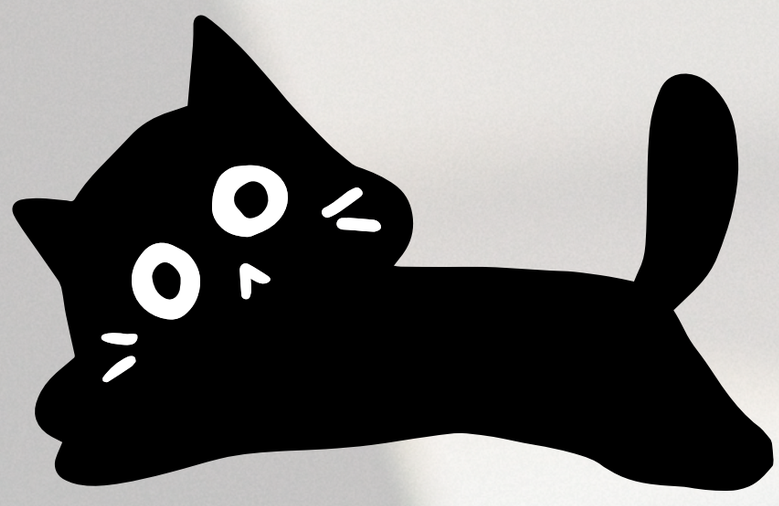
Persiaran Usahawan, Seksyen U1,

40150 Shah Alam Selangor

Telephone No.: +603 5163 4000

Fax No. : +603 5569 1903

# SERIES 1: C FUNDAMENTAL



EMY SATIRA AZRIN BINTI MOHAMED HAKKE

ILYA BINTI ISMAIL

NAAGAJOOTHI A/P ADIN NARAINA

# PREFACE

Welcome to the world of programming with C! Whether you're a complete beginner or a seasoned developer looking to reinforce your understanding, this eBook aims to provide a comprehensive guide to the fundamentals of the C programming language.

C is a powerful and versatile programming language that has been the foundation for countless software applications, operating systems, and embedded systems for decades. Its simplicity, efficiency, and portability make it an essential skill for anyone aspiring to become a proficient programmer.

In this eBook, we will start with the basics of C programming, covering essential concepts such as function input/output, variables, data types and control structures. Each topic will be explained in a clear and concise manner, with plenty of examples and exercises to help reinforce your understanding.

As you progress through the chapters, you will learn not only the syntax of the C language but also the underlying principles of computer programming. You will discover how to write clean, efficient, and maintainable code, and gain insight into best practices and common pitfalls to avoid.

Whether your goal is to develop software applications, explore systems programming, or dive into the world of embedded systems, mastering the fundamentals of C programming is an essential first step. This eBook is designed to be your companion on this exciting journey, providing you with the knowledge and skills you need to succeed.

**THANK YOU FOR YOUR SUPPORT.**
**WISH THIS EBOOK BENEFITS EVERYONE.**

# The Purr-fect Pros

**Mrs Emy Satira Azrin Binti Mohamed Hakke**
Lecturer of Diploma Medical Electronics Engineering
**emy@psa.edu.my**

B E  A  R E S P O N S I B L E  U S E R

Master of Technic And Vocational Education
Kolej Universiti Tun Hussein Onn (KUiTTHO)
Bsc Electrical Engineering (KUiTTHO)

**Mrs Ilya binti Ismail**
Lecturer of Diploma Communication Electronics Engineering
**ilya@psa.edu.my**

B E  S M A R T  T O  Y O U R  D E V I C E S

Master of Science in
Telecommunication and Information Engineering,
Universiti Teknologi MARA(UiTM)
Bachelor of Electrical Engineering(Telecommunication)
Universiti Teknologi Malaysia (UTM)

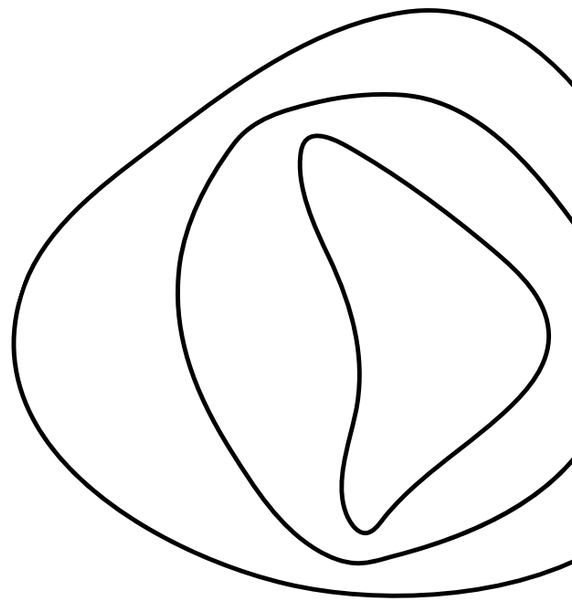
**Mrs Naagajoothi A/P Adin Naraina**
Lecturer of Diploma Medical Electronics Engineering
**naagajothi@psa.edu.my**

B E  A  R E S P O N S I B L E  U S E R

Diploma in Electrical Electronic Engineering (with Education)
Institut Teknologi Tun Hussein Onn (ITTHO)
Bachelor in Electrical Electronic Engineering (UPM)

# TABLE OF CONTENTS

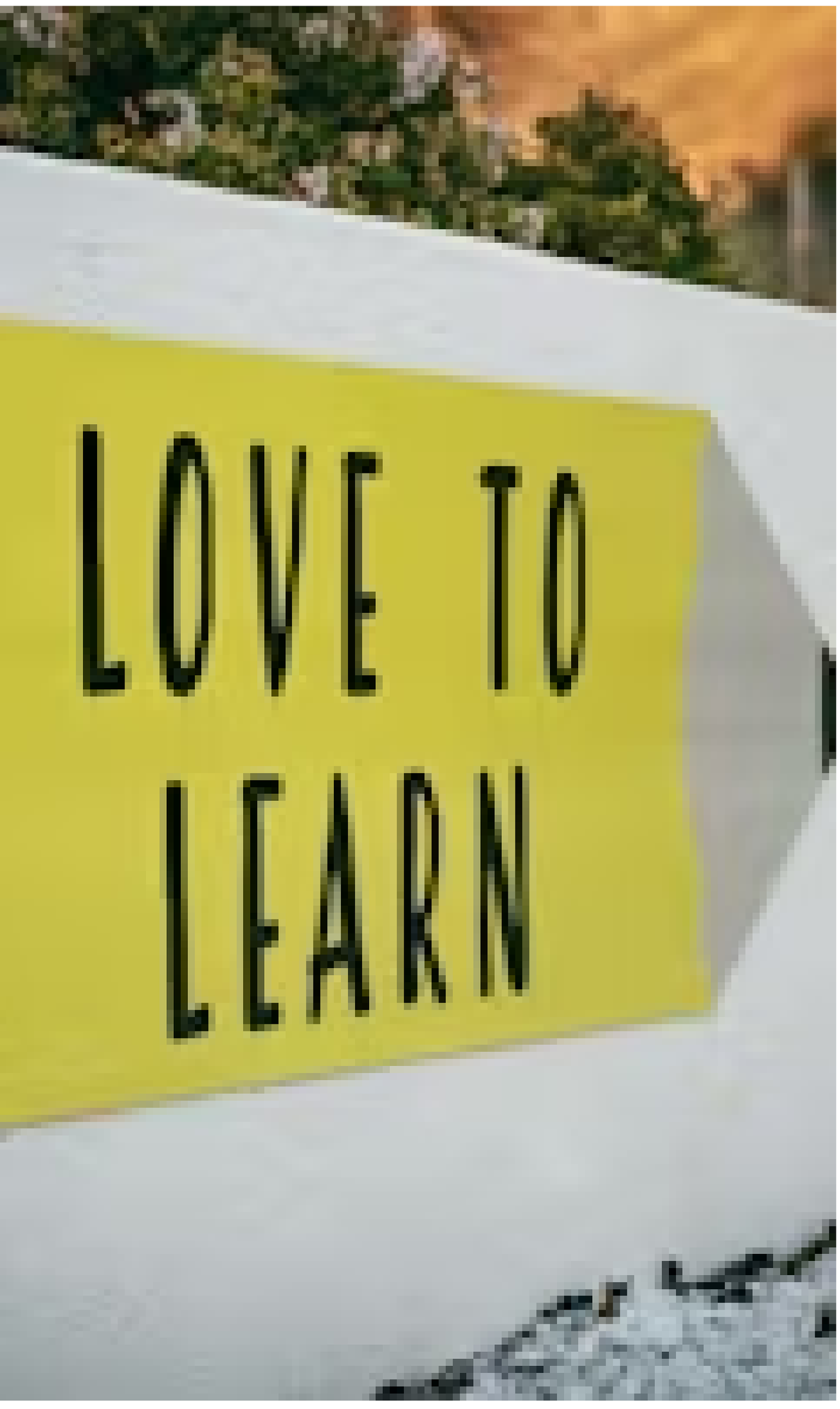




# INTRODUCTION to C PROGRAMMING

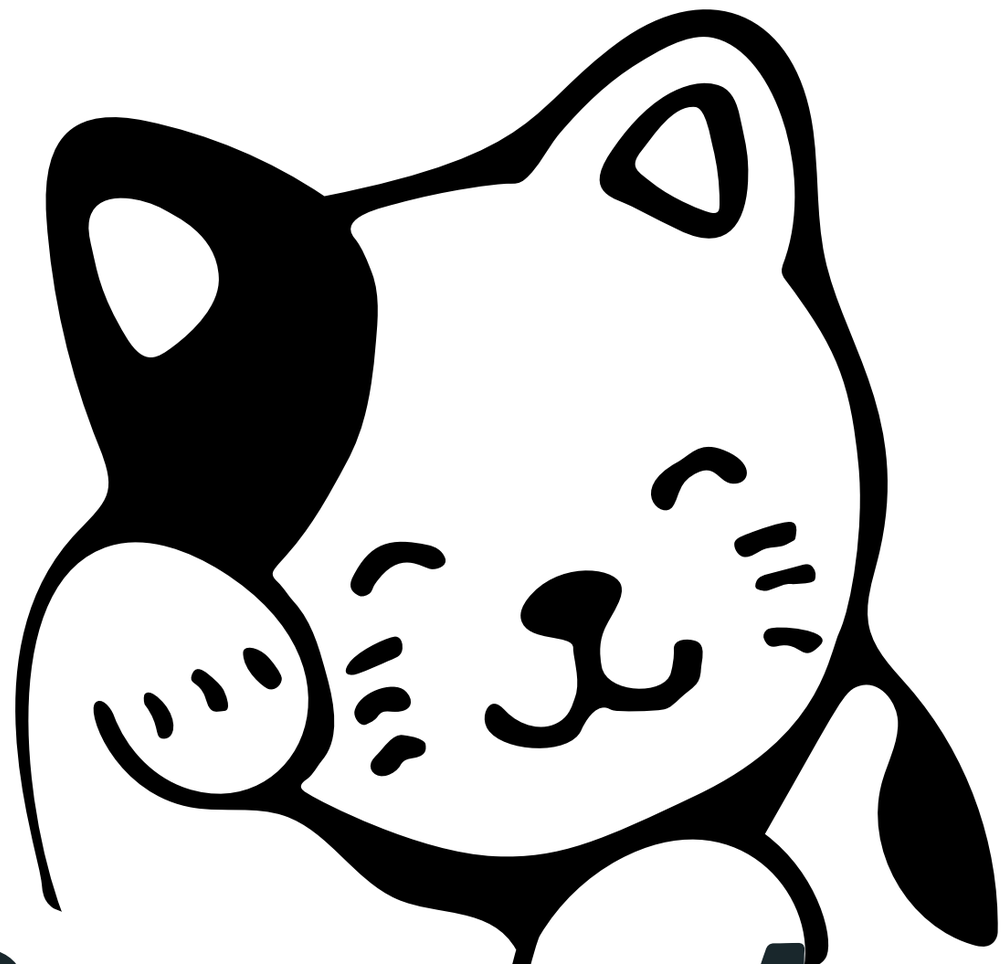
# INTRODUCTION

C is a programming language that allows a software engineer to efficiently communicate with a computer




It's been used for a wide variety of programs including firmware for micro-controllers, operating systems, applications, and graphics programming

# SCAN FOR MORE INFO



# Be friends with C

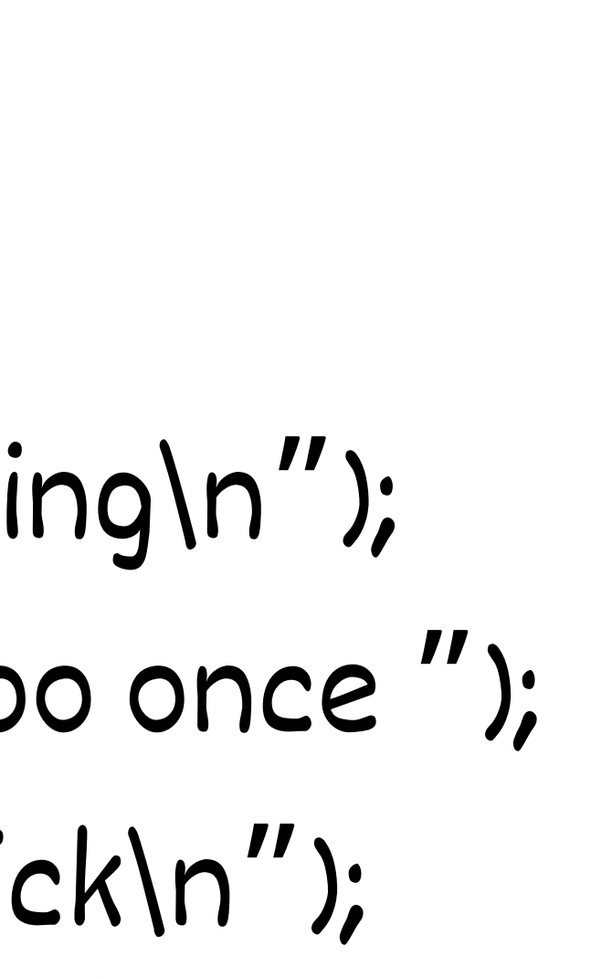
# Be friends with C

## C Program Structure

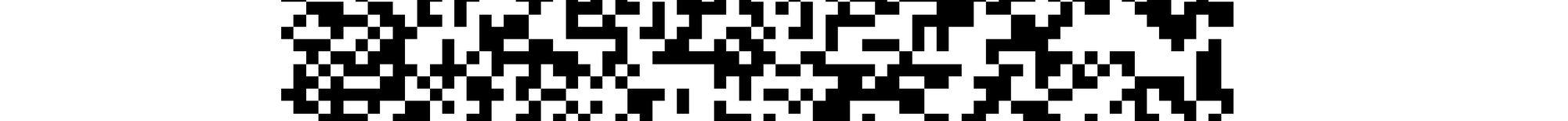
An example of simple program in C

```
#include <stdio.h>
int main(void)
{
    printf("I love programming\n");
     printf("You will love it too once ");
     printf("you know the trick\n");
return 0;
}
```

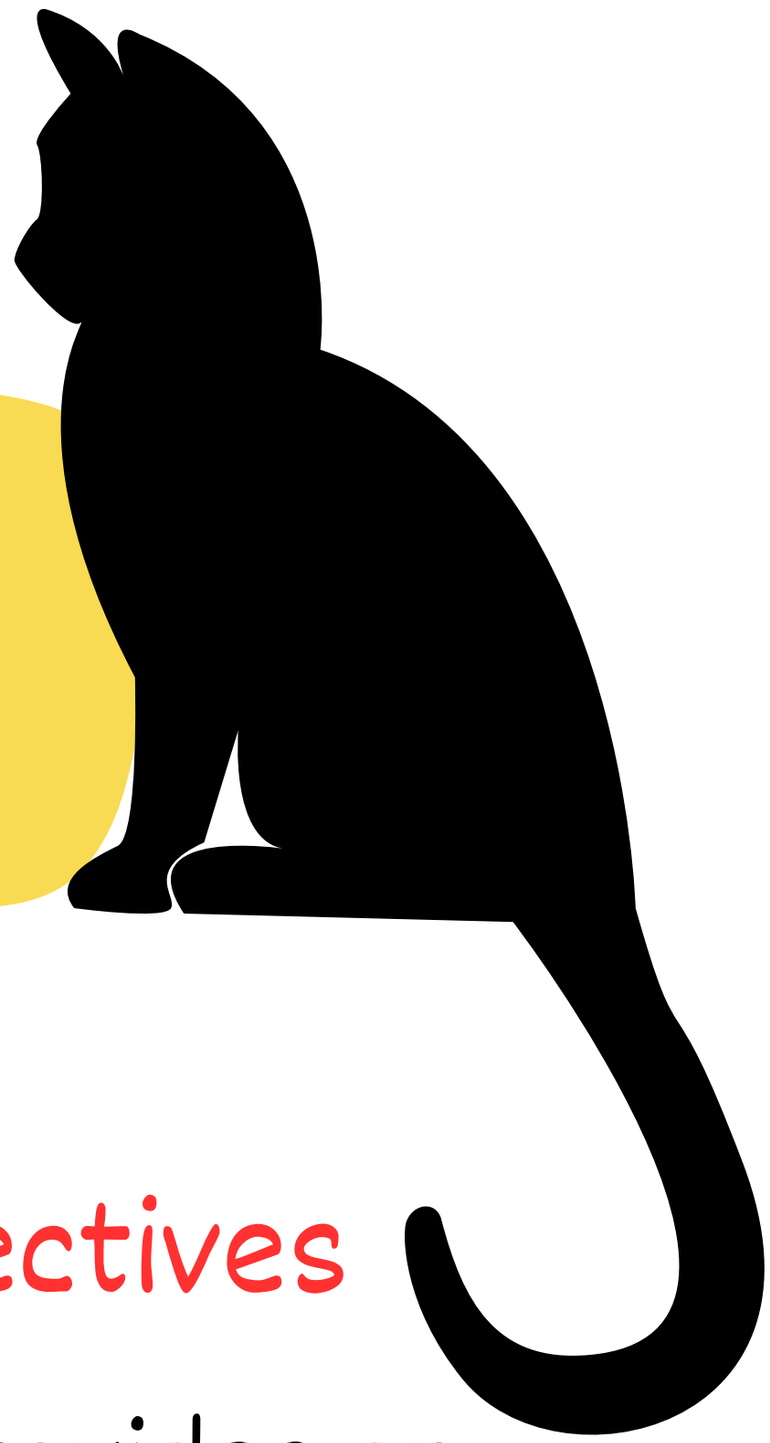




# SCAN FOR MORE INFO



# Be friends with C



## Function Preprocessor Directives

- A C program line begins with # provides an instruction to the C preprocessor presentation

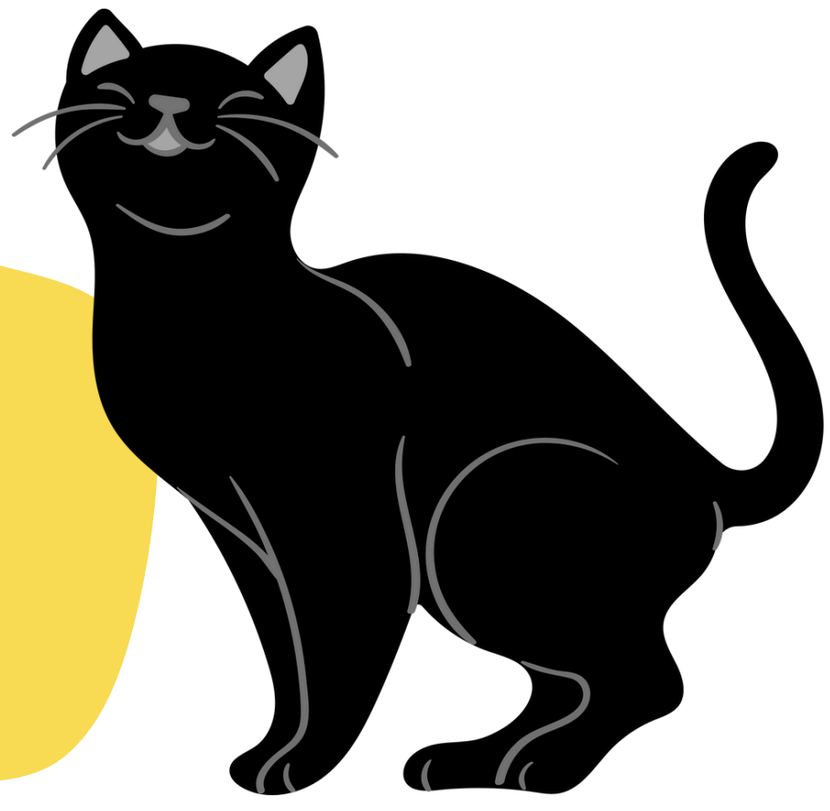
- In example (#include<stdio.h>) identifies the header file for standard input and output needed by the printf().

- #include -- includes contents of a named file. Files usually called header files.

#include <math.h> -- standard library math file.

#include <stdio.h> -- standard library I/O file

# Be friends with C

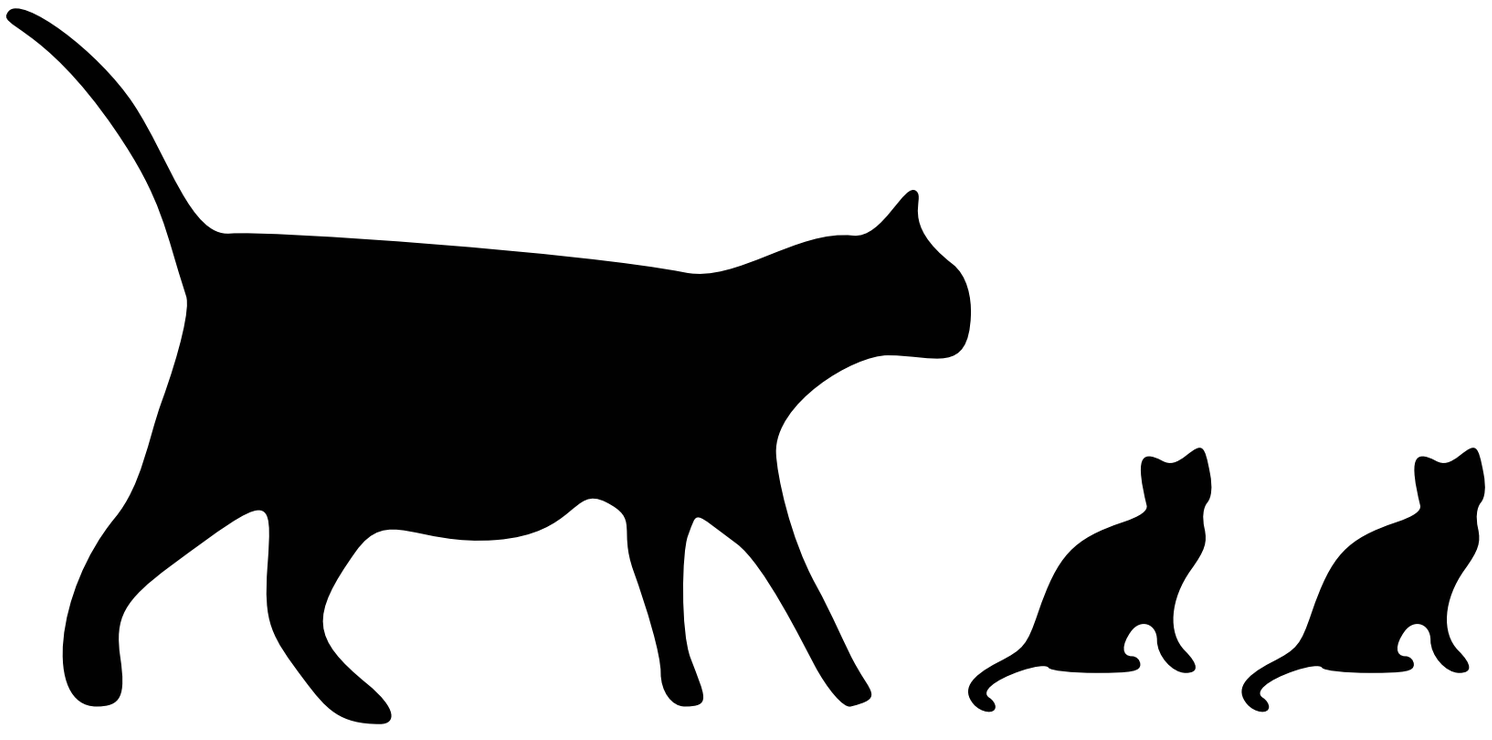


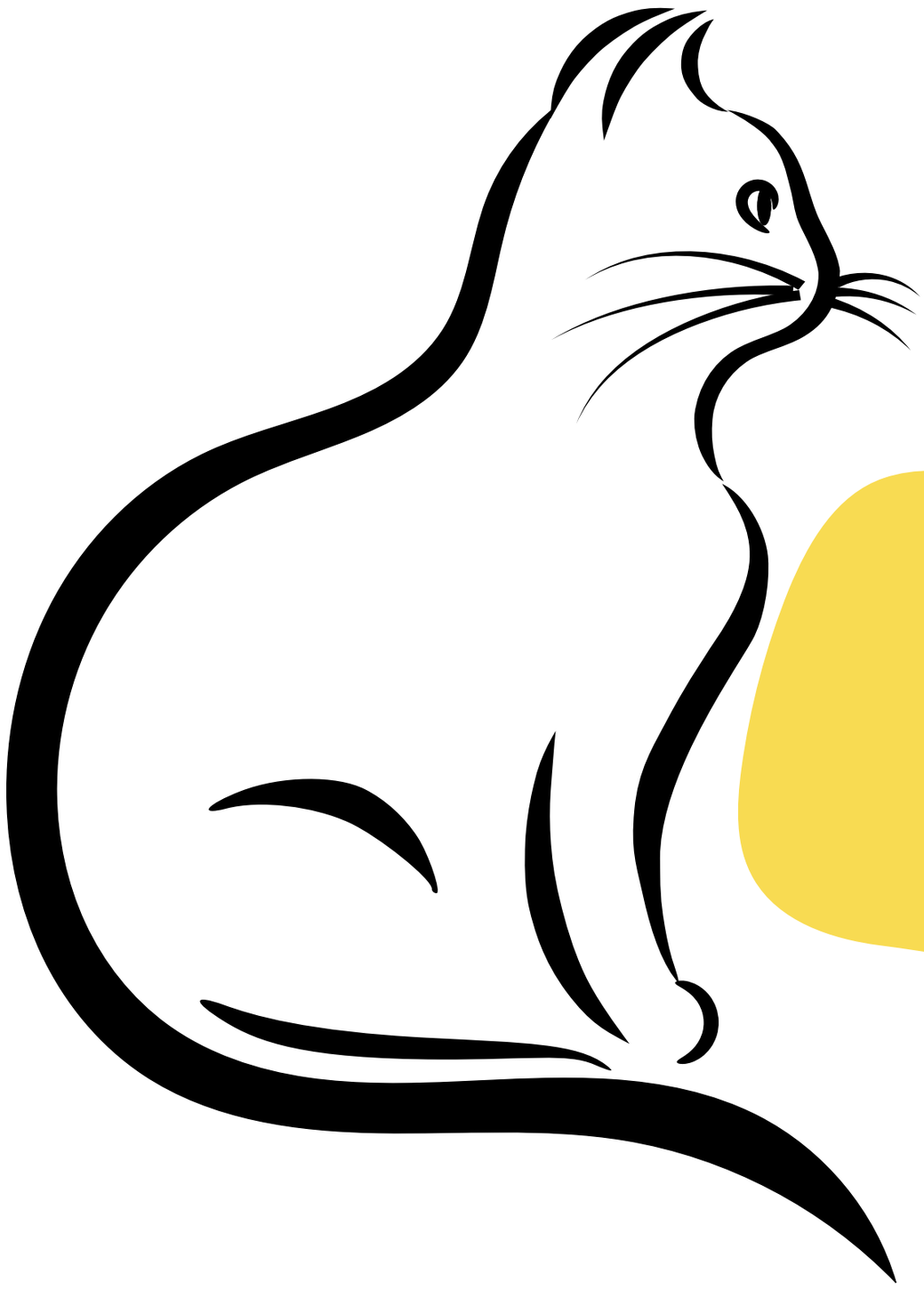
## Function Printf

The printf function is a part of the C standard library <stdio.h> and it can allow formatting the output in numerous ways.

## Syntax of printf

```
printf ( "formatted_string", arguments_list);
```



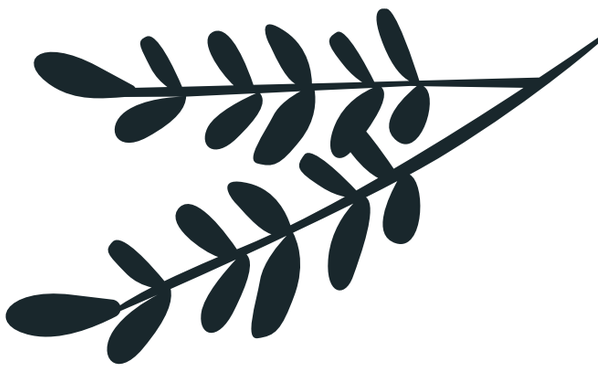


Be friends with C

# C Variables

- a memory location with some name that helps store some form of data and retrieves it when required.

- We can store different types of data in the variable and reuse the same variable for storing some other data any number of times.



# Be friends with C

## Function Scanf

- Stands for Scan Formatted String. It is used to read data from stdin (standard input stream i.e. usually keyboard) and then writes the result into the given arguments.

- It accepts character, string, and numeric data from the user using standard input.

- scanf also uses format specifiers like printf.

- Data types for scanf

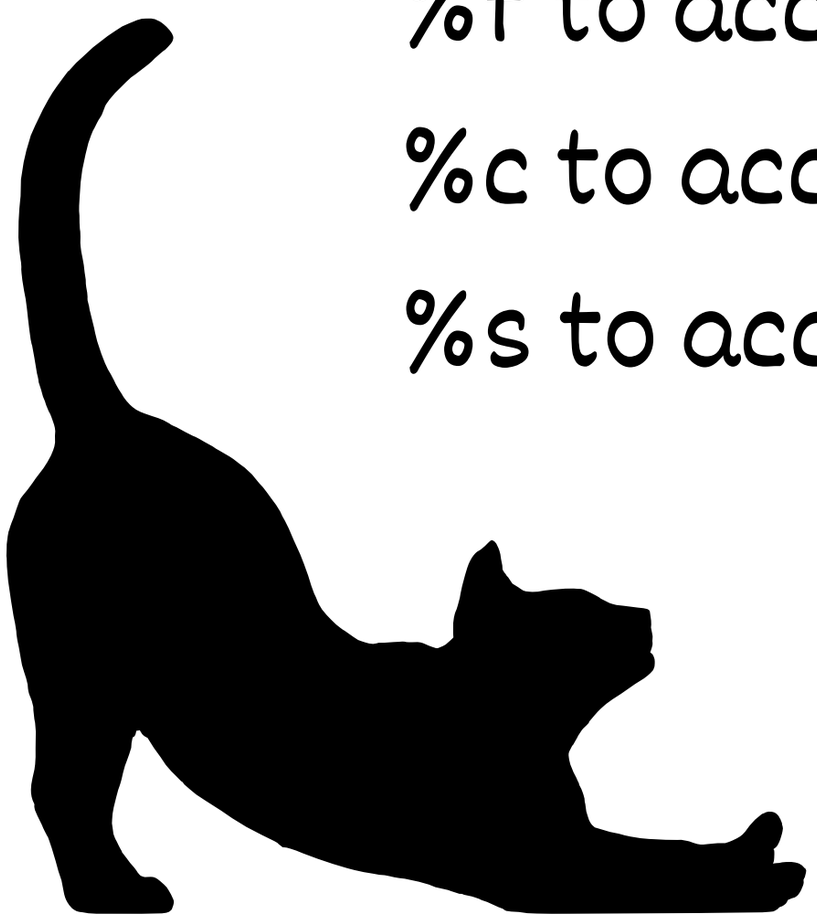
  %d to accept input of integers. - int

  %lf to accept input of long integers. - double

  %f to accept input of floating point. - float

  %c to accept input of character types. - char

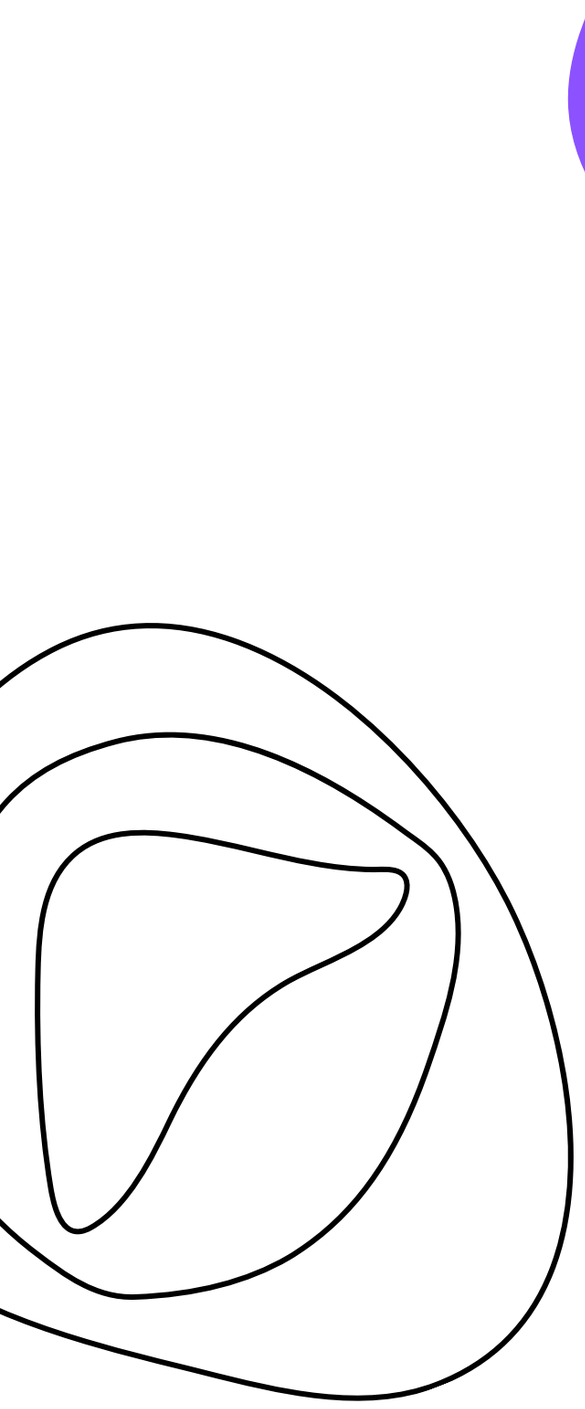
  %s to accept input of a string. - char



# Lets Do It

# C Output

## Printf()



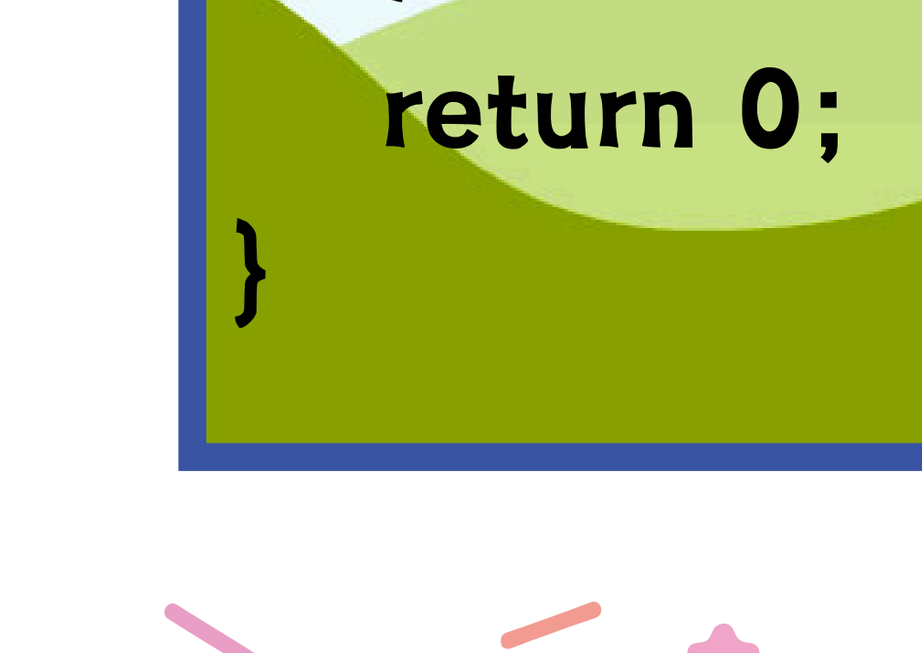
# Lets Do It

## C Output

```
#include <stdio.h>
int main()
{
    // Displays the string inside quotations

    printf("C Programming");
    return 0;
}
```

Output

C Programming

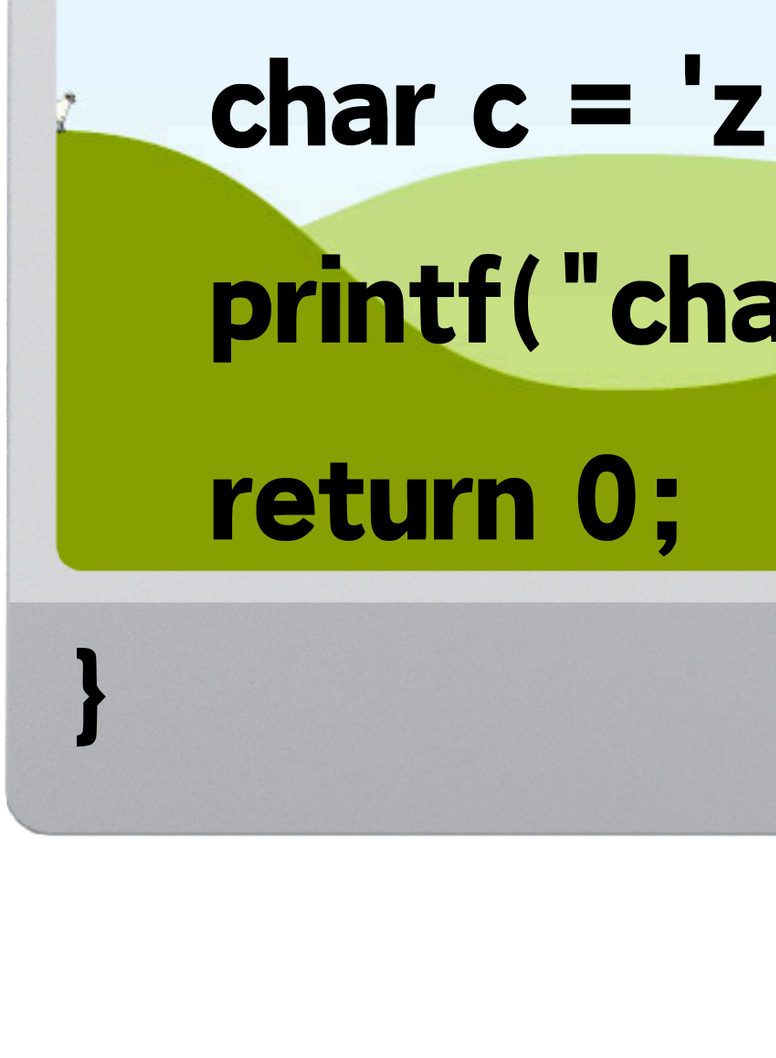


# Lets Do It

## Print Characters

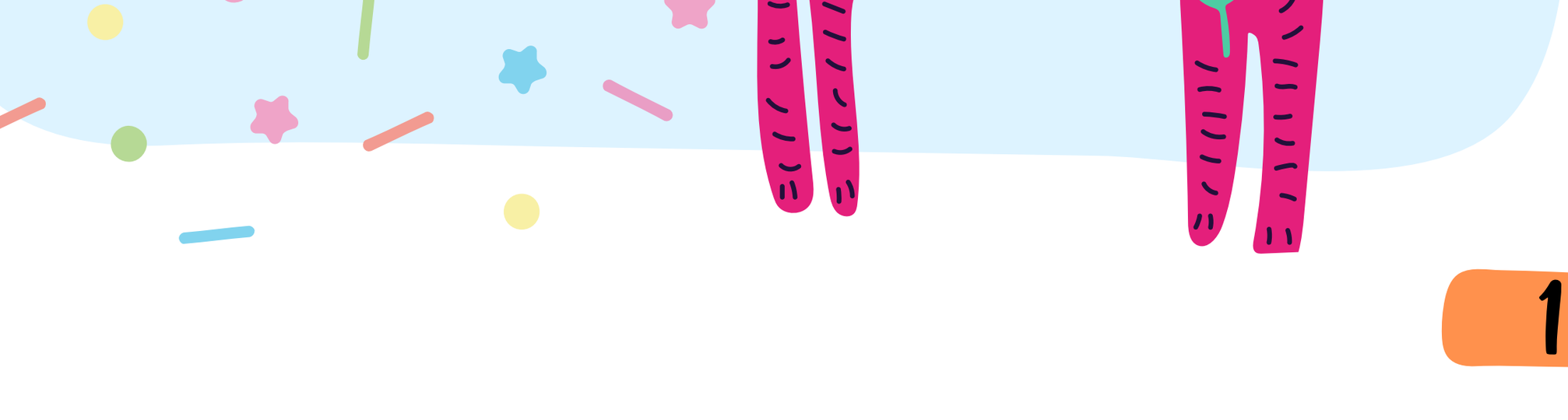
```
#include <stdio.h>
int main()
{
    char c = 'z';
    printf("character = %c", c);
    return 0;
}
```

## Output



```
character = z
```

# SCAN FOR MORE INFO



# Lets Do It

## C Output

```
#include <stdio.h>
main()
{
printf("You are learning printf() function\n");
printf("You are learning printf() function");
}
```

## Output

```
You are learning printf() function
You are learning printf() function
```

# Lets Do It

## Integer Output

```
#include <stdio.h>

int main()

{

    int testInteger = 21;

    printf("Number = %d", testInteger);

    return 0;

}
```

Output

```
Number = 21
```

# Lets Do It

## Integer Output

```
#include <stdio.h>

int main()

{

    int NOM = 60;

    printf("Number = %d", NOM);

    return 0;

}
```

Output

```
Number = 60
```

# Lets Do It

## Float and Double Output

```
#include <stdio.h>

int main()
{
    float numb1 = 20.1;
    double numb2 = 14.2;


    printf("number1 = %f\n", numb1);
    printf("number2 = %lf", numb2);
    return 0;
}
```

Output

```
number1 = 20.100000
number2 = 14.200000
```

# Lets Do It

## Float and Double Output

```
#include <stdio.h>

int main()
{
    float numb1 = 23.231355;
    double numb2 = 30.291632;


    printf("number1 = %.2f\n", numb1);
    printf("number2 = %.3lf", numb2);
    return 0;
}
```

Output.

```
number1 = 23.23
number2 = 30.292
```

# Lets Do It

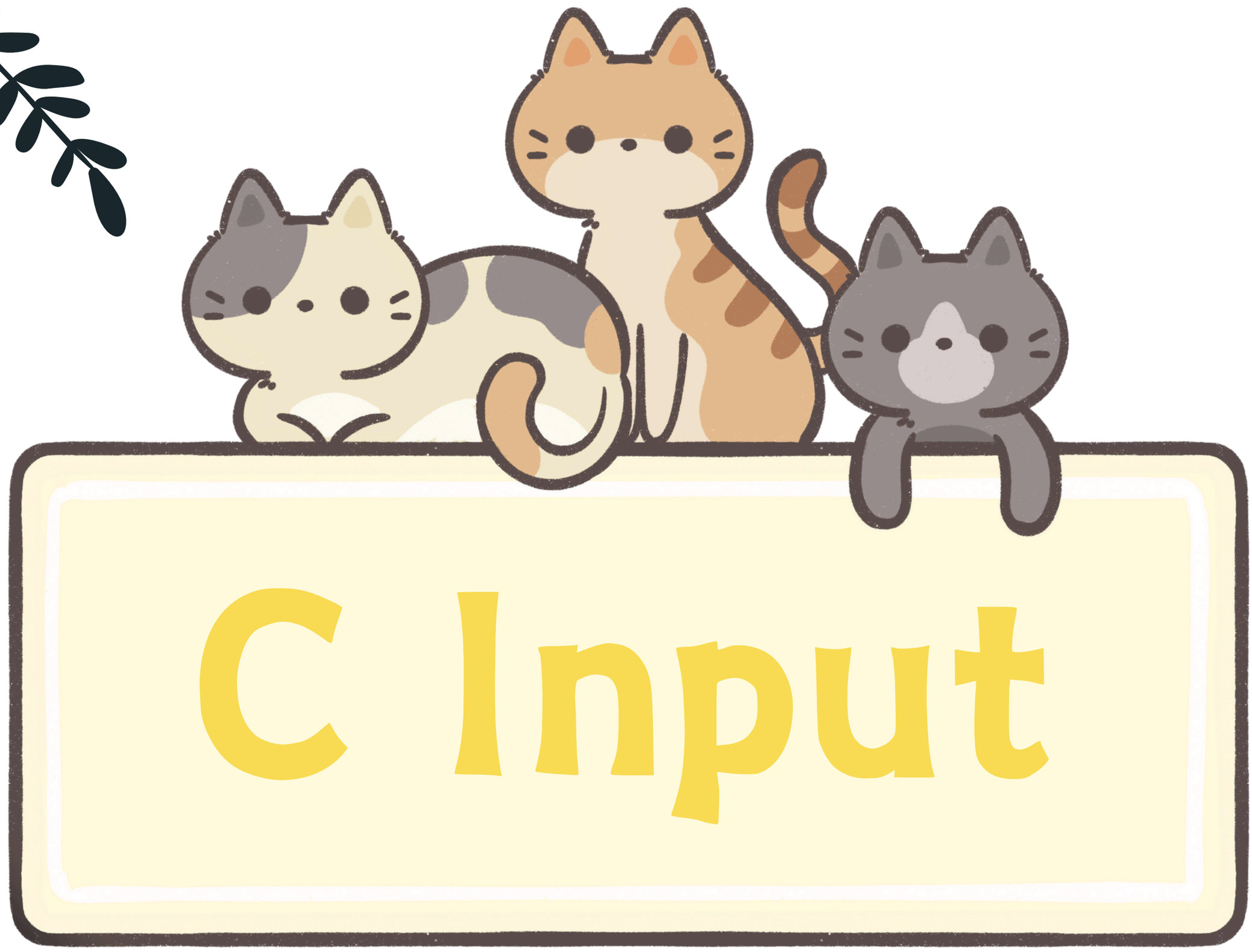
## Print a Variable using Specifier

```
#include <stdio.h>
int main()
{    int num1 = 99;
     int num2 = 1;
 printf("The sum of %d and %d is %d\n",
num1, num2, num1 + num2);
return 0;}
```

Output

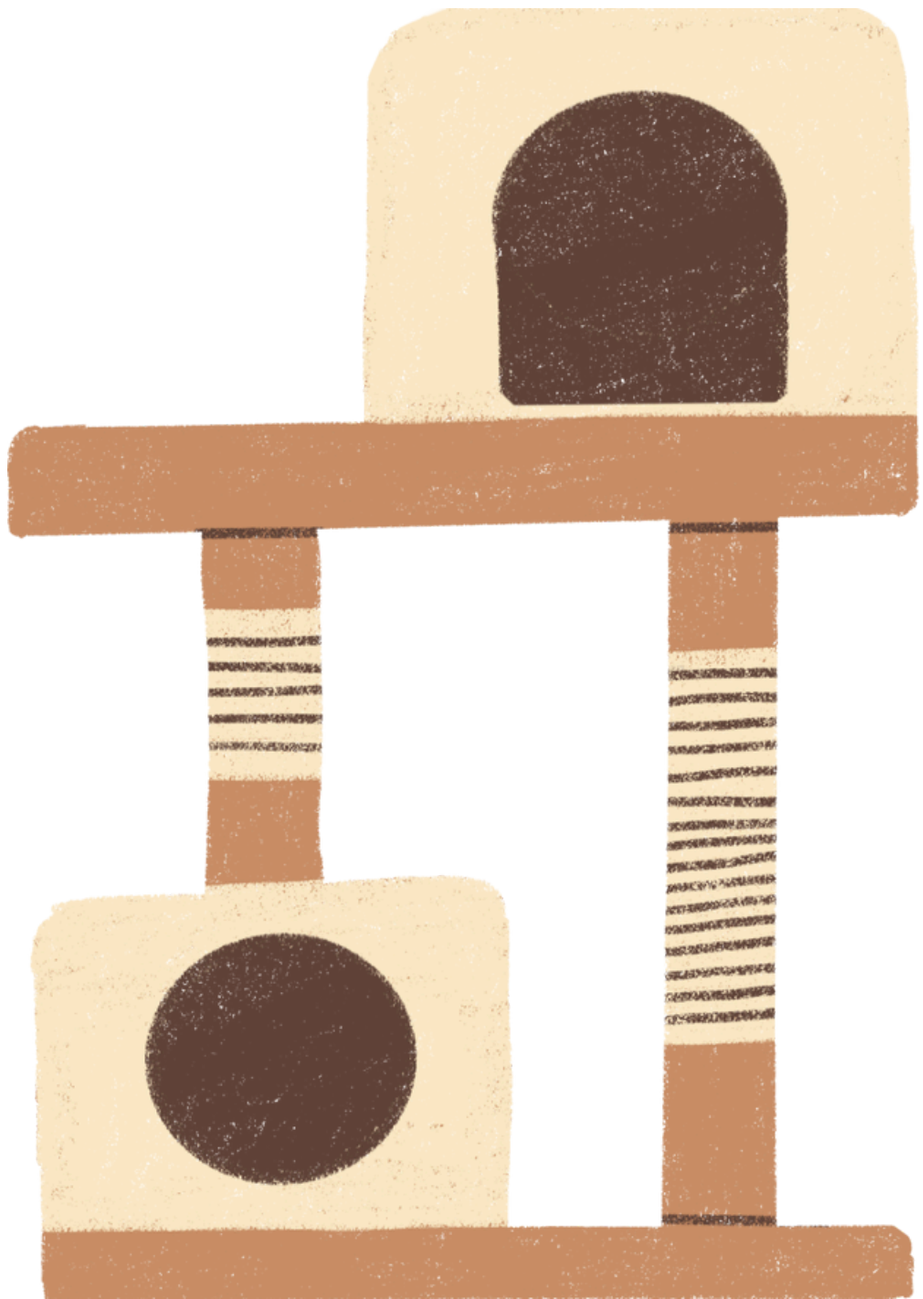
The sum of 99 and 1 is 100

# Lets Do It



# C Input

## scanf()

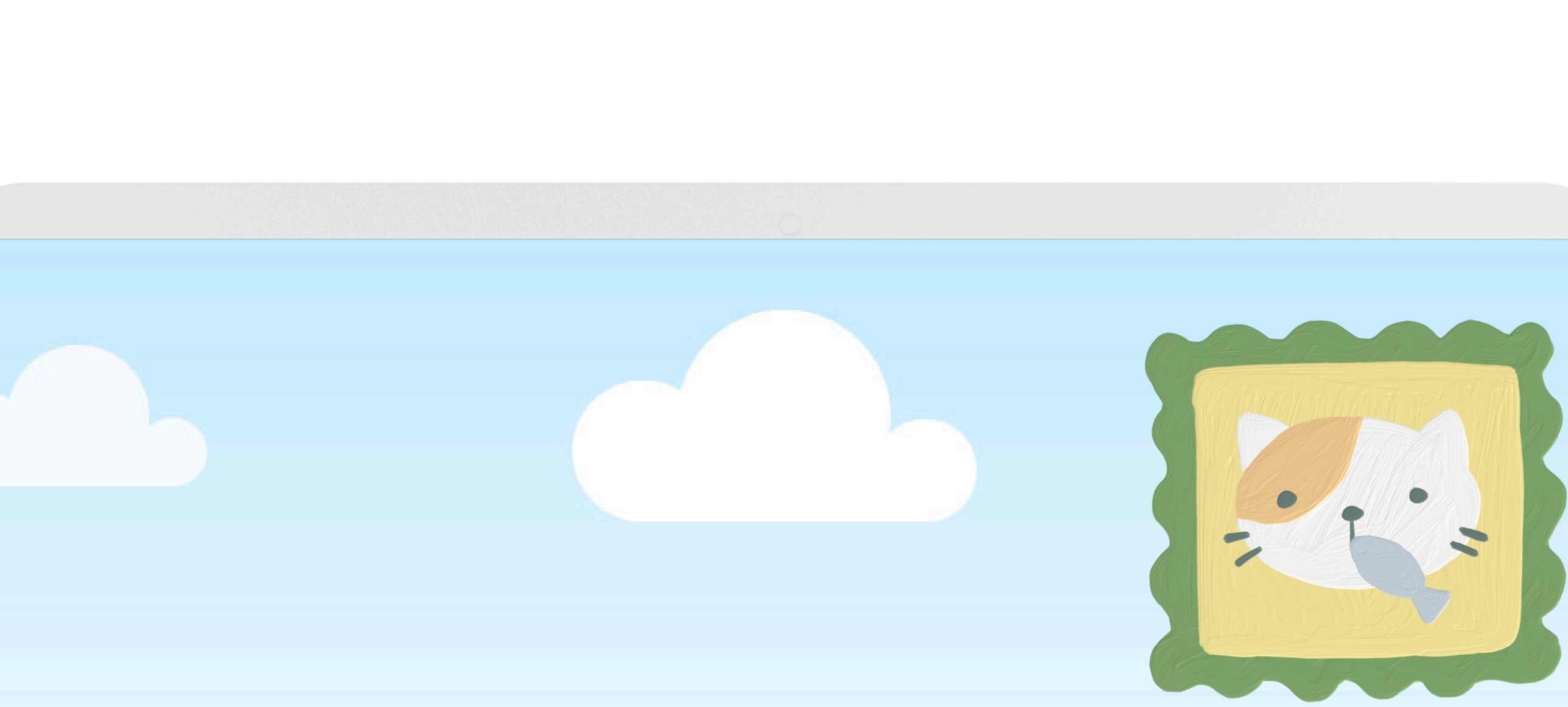


# C Input

## Integer Input/Output

```
#include <stdio.h>
int main()
{
    int testInteger;
    printf("Enter an integer: ");
    scanf("%d", &testInteger);
    printf("Number = %d",testInteger);
    return 0;
}
```

# Output



Lets Do It

# C Input

## Integer Input/Output

```
#include <stdio.h>
int main()
{
    int NOM;
    printf("Enter an integer: ");
    scanf("%d", &NOM);
    printf("Number = %d",NOM);
    return 0;
}
```

# Output

```
Enter an integer: 30

Number = 30
```

Lets Do It


# C Input

## Float and Double Input/Output

```
#include <stdio.h>
int main()
{
    float nom1;
    double nom2;
    printf("Enter a number: ");
    scanf("%f", &nom1);
    printf("Enter another number: ");
    scanf("%lf", &nom2);
    printf("nom1 = %f\n", nom1);
    printf("nom2 = %lf", nom2);
    return 0;
}
```

# Lets Do It

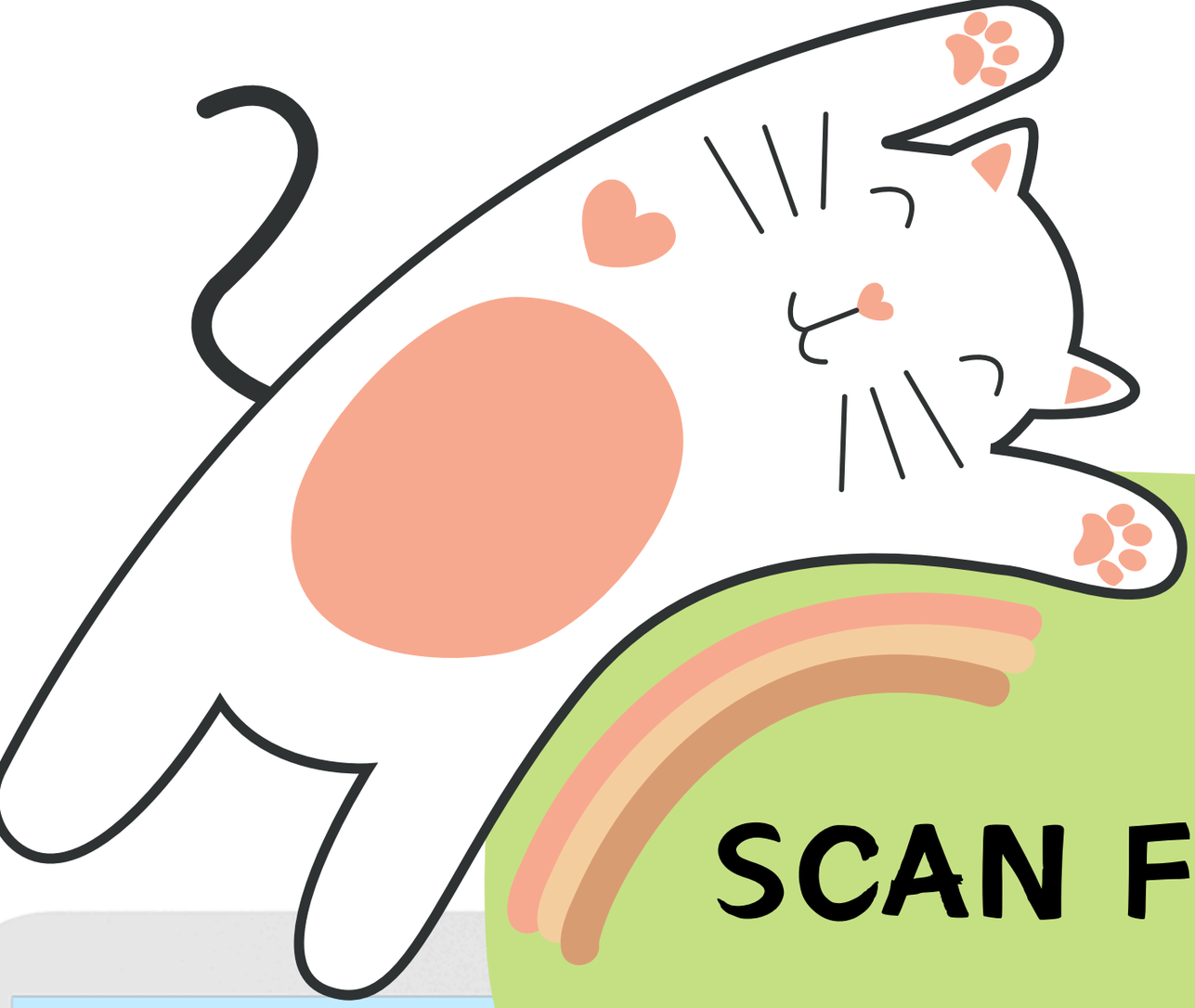
## Output

Enter a number: 12.345

Enter another number: 6.7

num1 = 12.345000

num2 = 6.700000

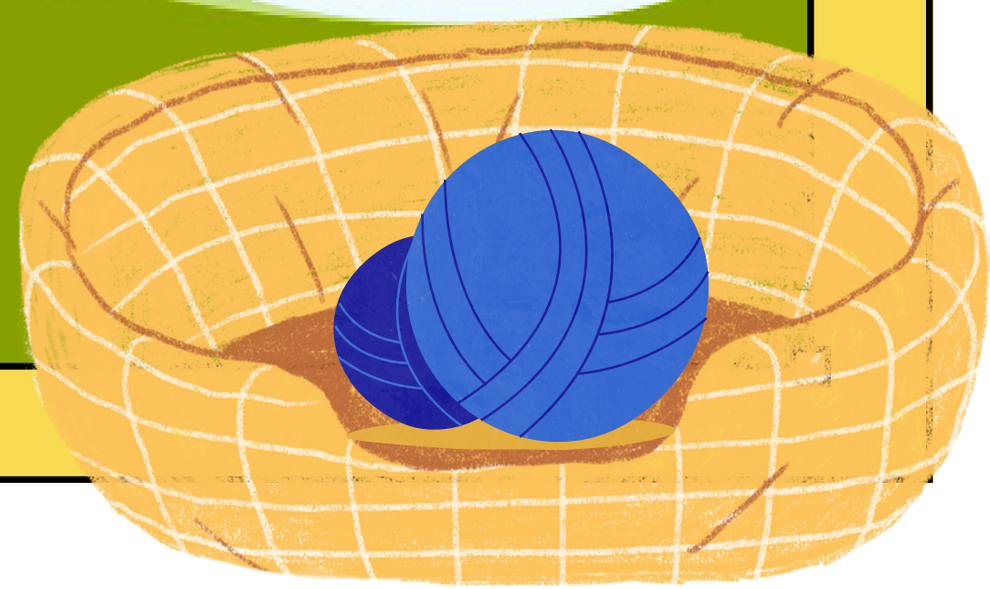


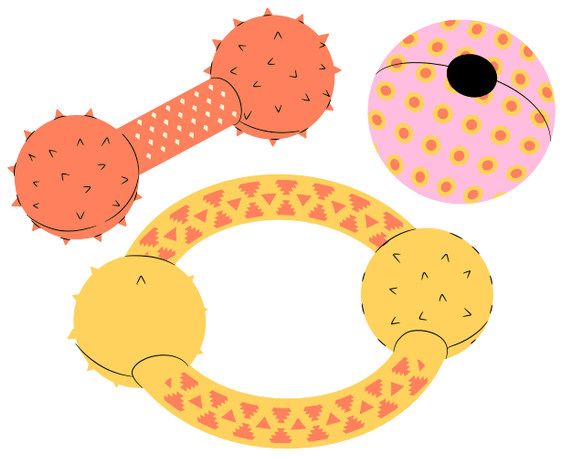
# SCAN FOR MORE INFO

# C Input

## C Character Input Output

```
#include <stdio.h>
int main()
{
    char c;
    printf("Enter a character: ");
    scanf("%c",&chr);
    printf("You entered %c.", c);
    return 0;
}
```





### Output

```
Enter a character: e

You entered e
```

# C Input

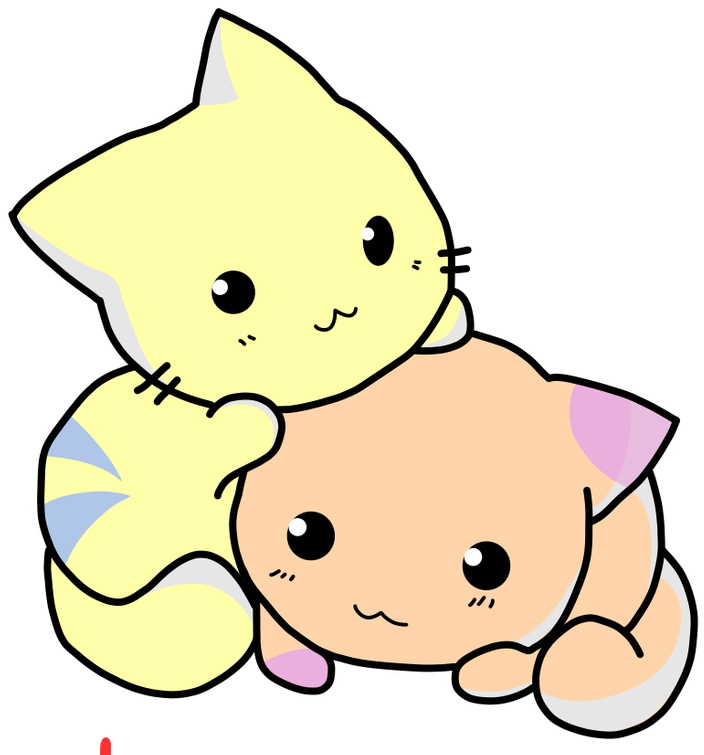
## C Character Input Output

```
#include <stdio.h>
int main()
{
    char s [30];
    printf("Enter your name: ");
    scanf("%s",&s);
    printf("Your name is %s.", s);
    return 0;
}
```

### Output

```
Enter your name: mohd
Your name is mohd.
```

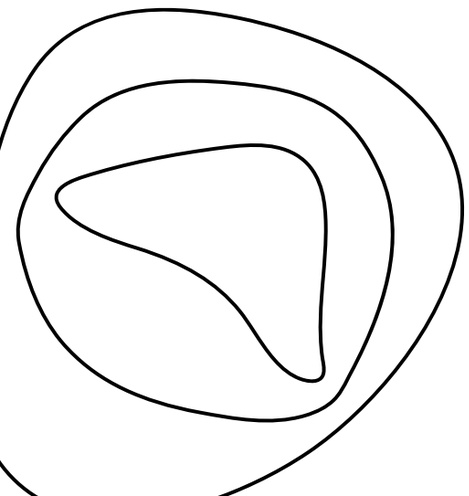
Lets Do It



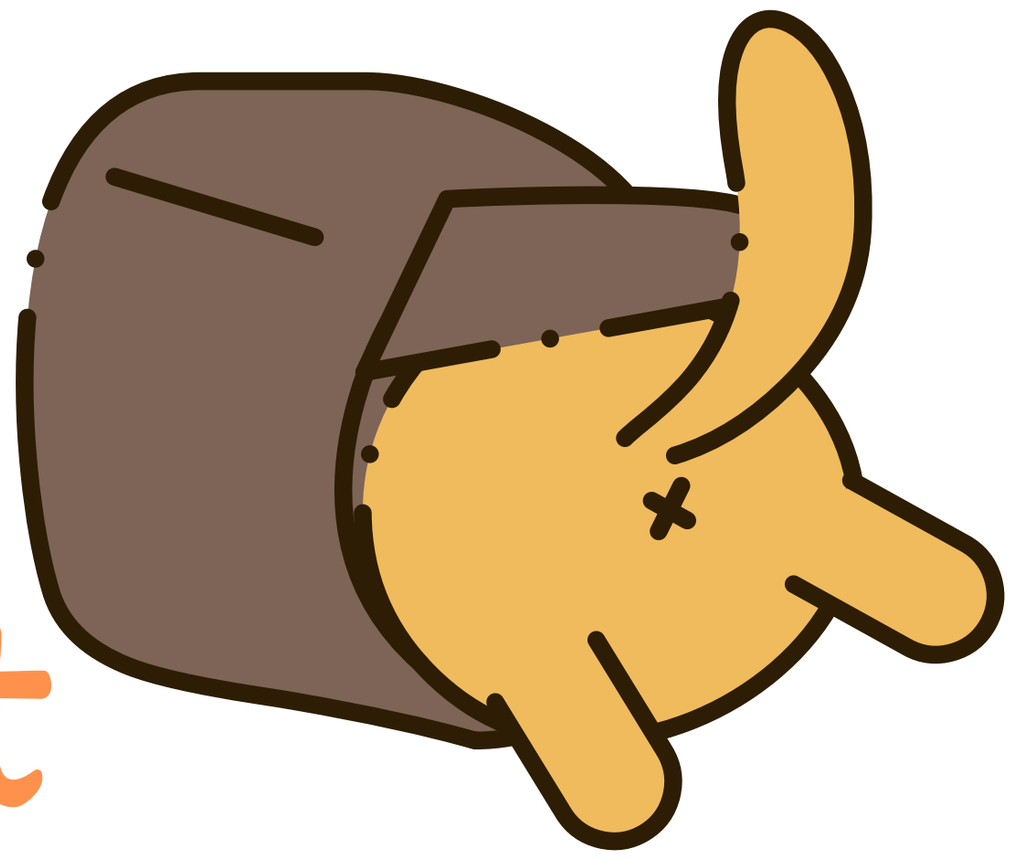
# C Input

## Input Output Multiple Values

```
#include <stdio.h>
int main()
{
    char x;
    float y;

    printf("Enter character and then a float: ");

// Taking multiple inputs

    scanf("%c%f", &x, &y);
    printf("You entered %c and %.2f", x, y);
    return 0;
}
```

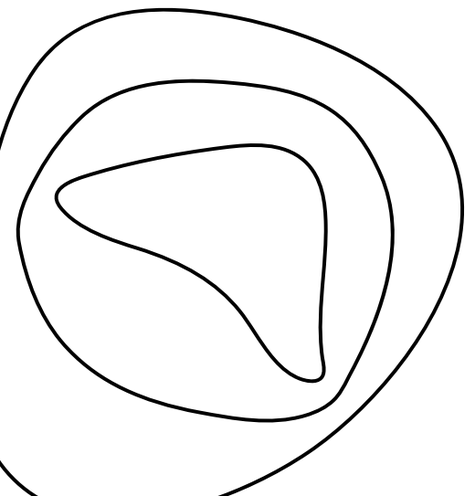


# Lets Do It

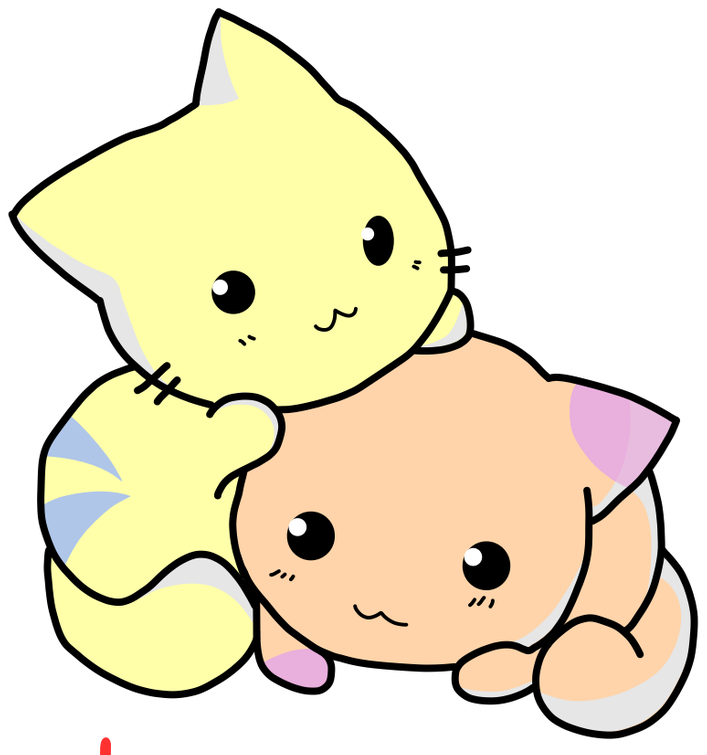


## Output

Enter character and then a float:
A 67.45678
You entered A and 67.46



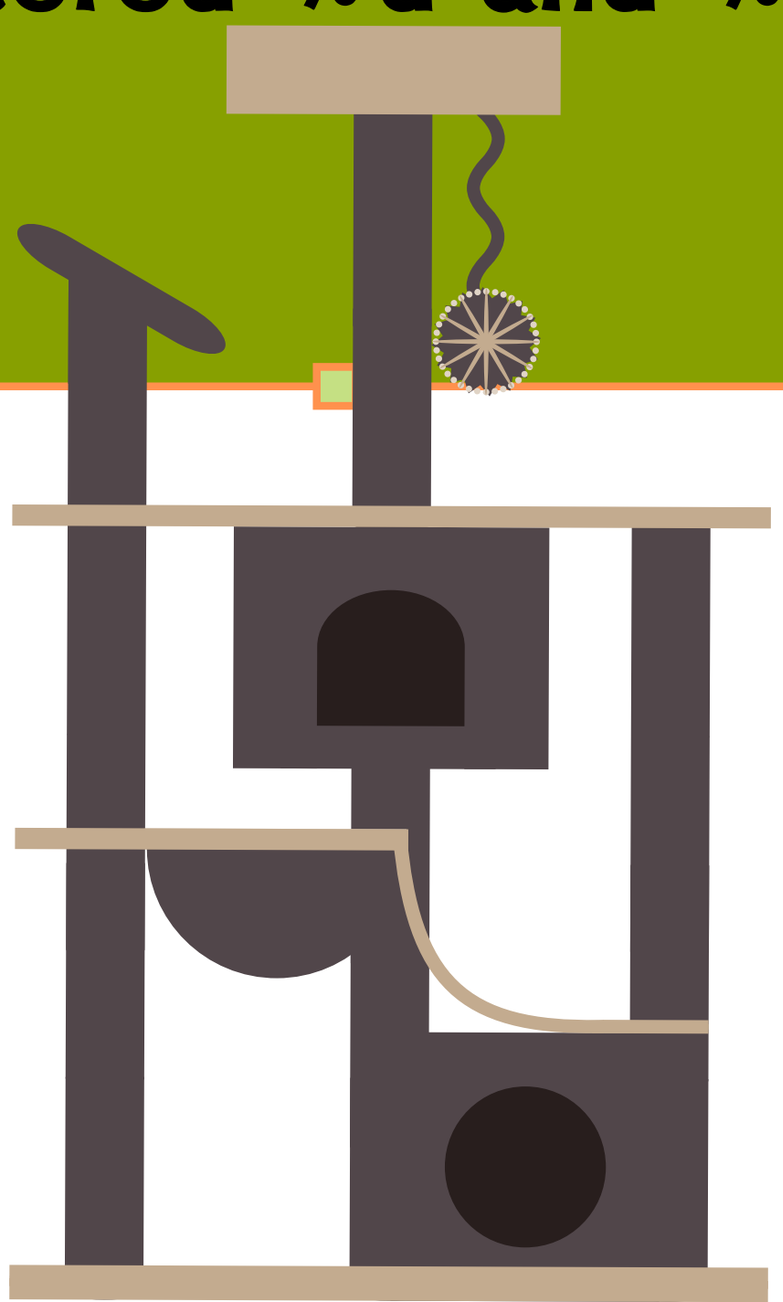
# Lets Do It



## C Input

### Input Output Multiple Values

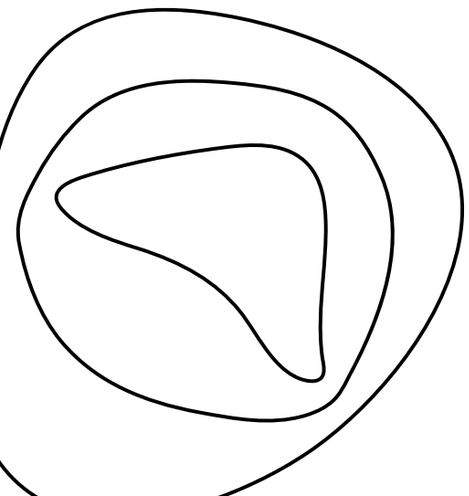
```
#include <stdio.h>
int main()
{
    int x;
    float y;

    printf("Enter integer and then a float: ");

// Taking multiple inputs

    scanf("%d%f", &x, &y);
    printf("You entered %d and %f", x, y);
    return 0;
}
```



# Lets Do It

## Output

```
Enter integer and then a float:
-10      7.5
You entered -10 and 7.500000
```



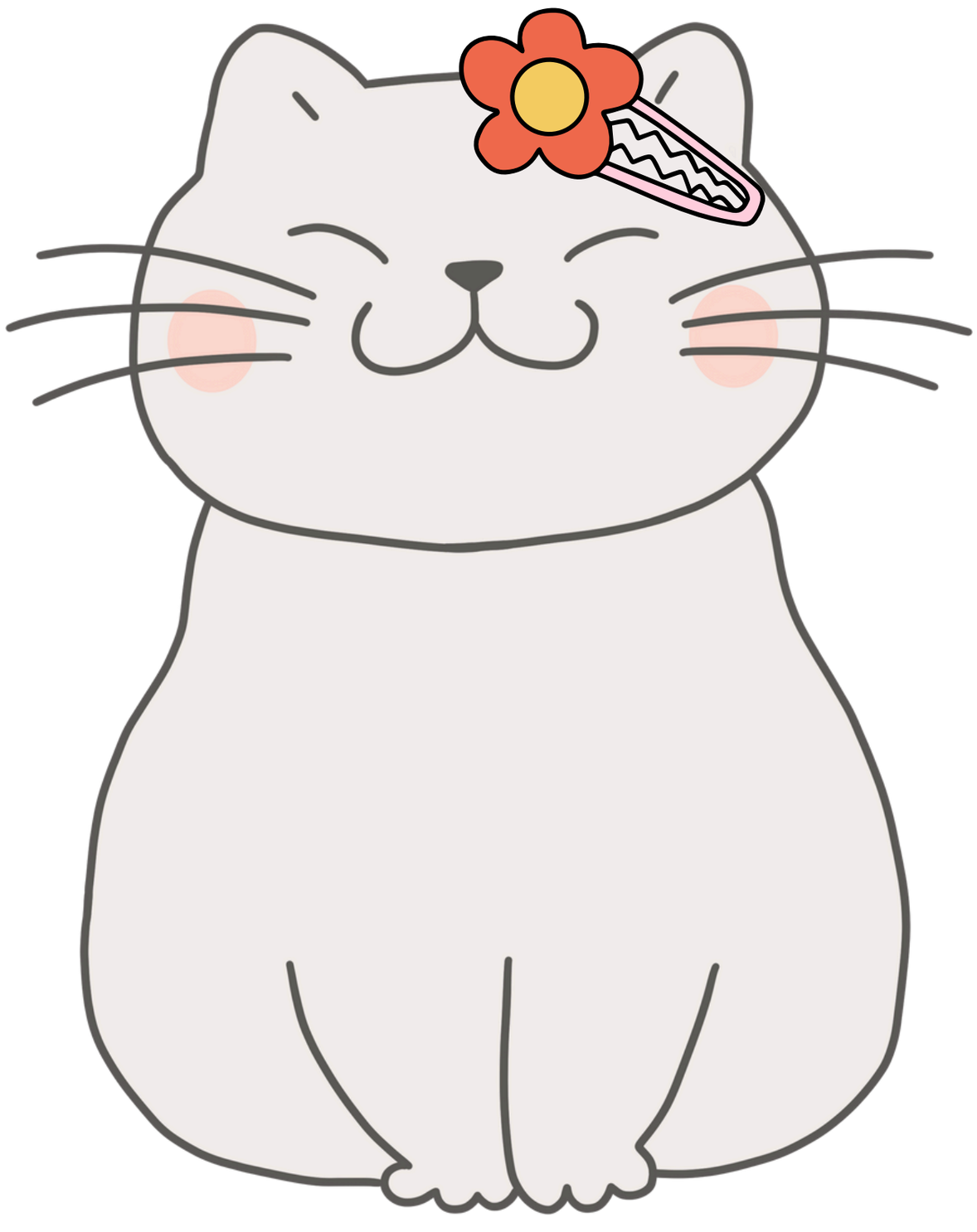
# SCAN FOR MORE INFO

# LET'S DO EXERCISES

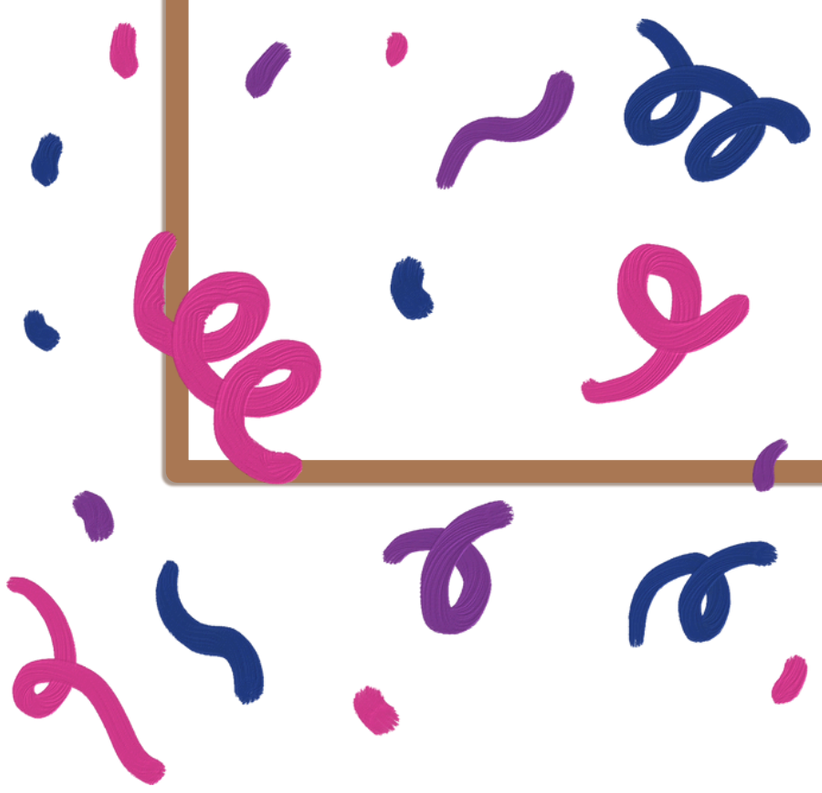
# EXERCISES

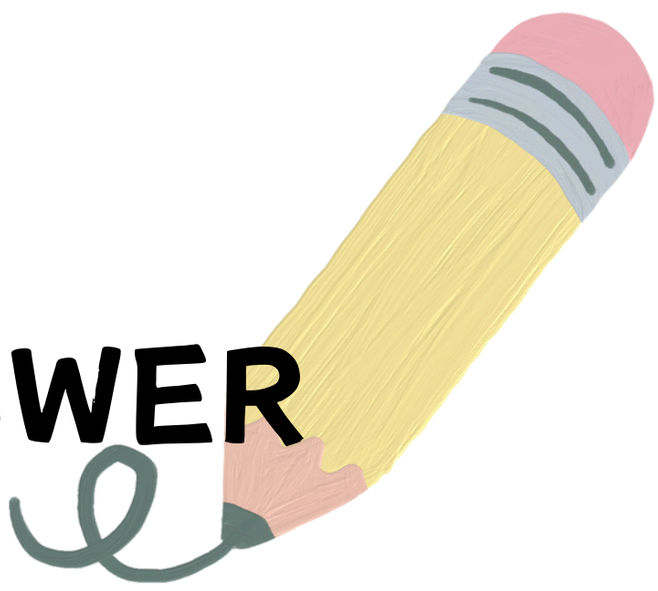
## QUESTION 1:

**Write a Program to find the user's age.**

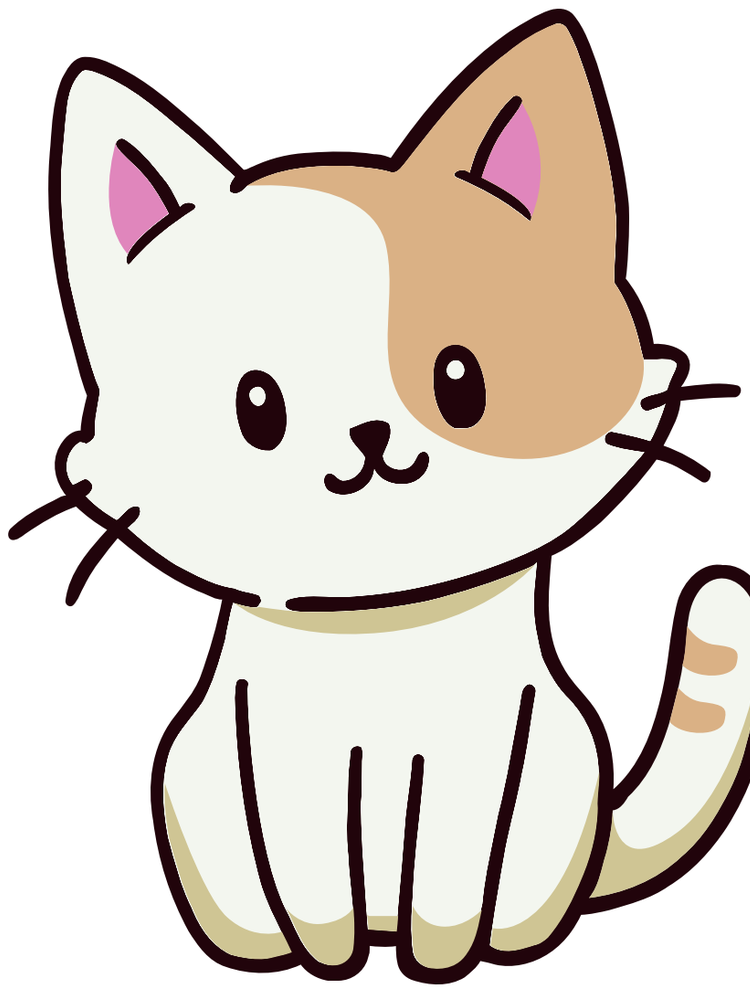


# QR ANSWER





## SCAN THE QR FOR ANSWER

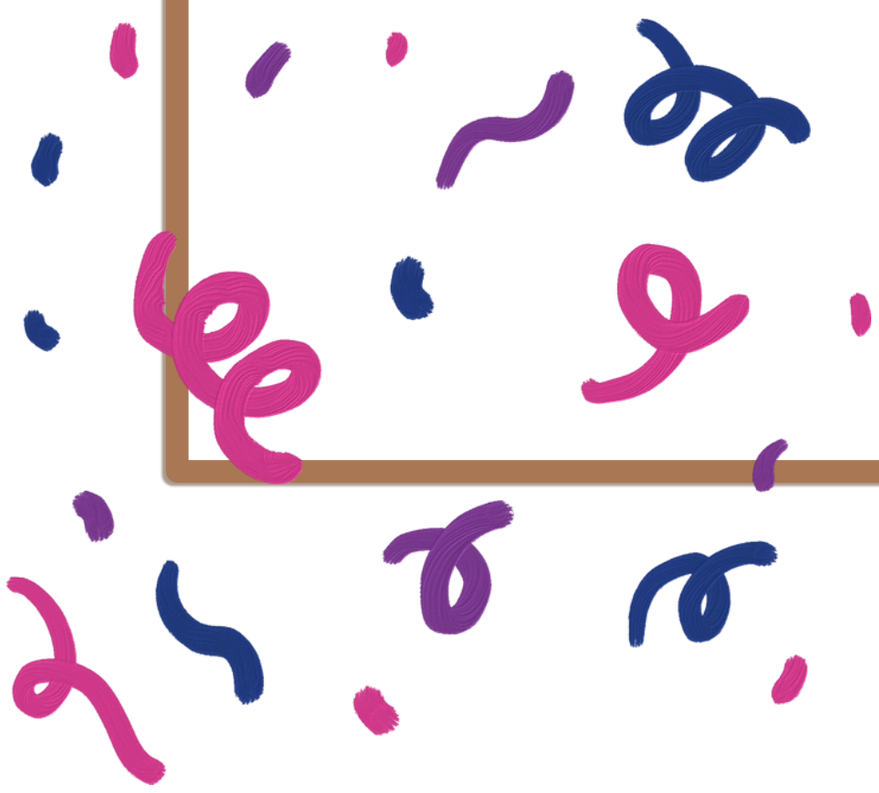


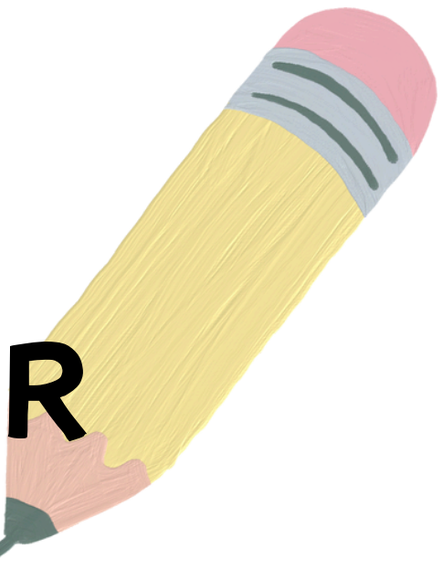
# EXERCISES

## QUESTION 2:

**Write a Program to find the Sum of two numbers entered by the user.**

# QR ANSWER





## SCAN THE QR FOR ANSWER



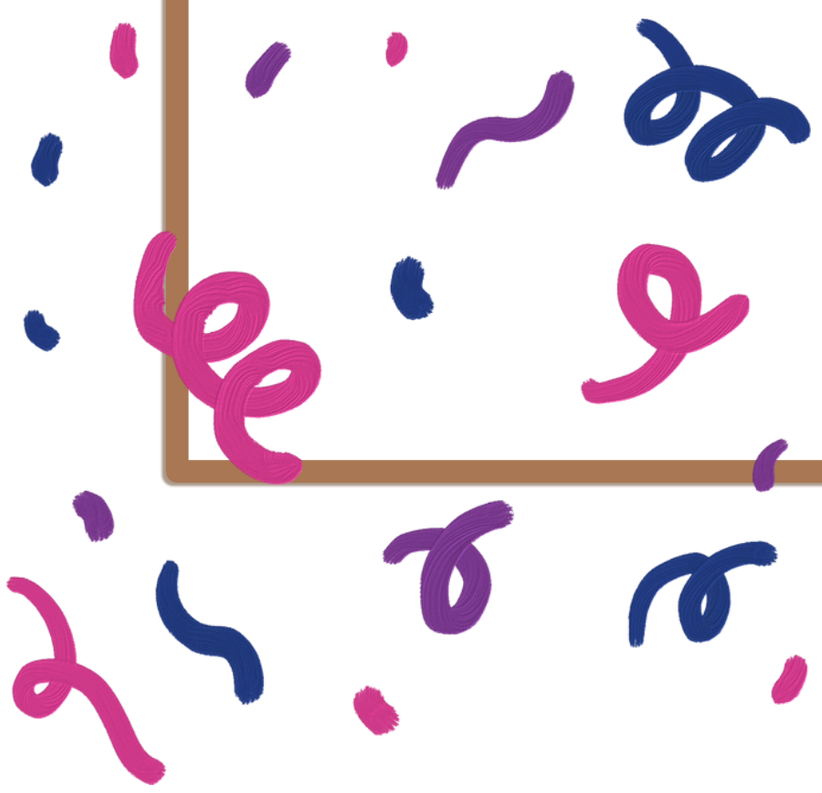
# EXERCISES

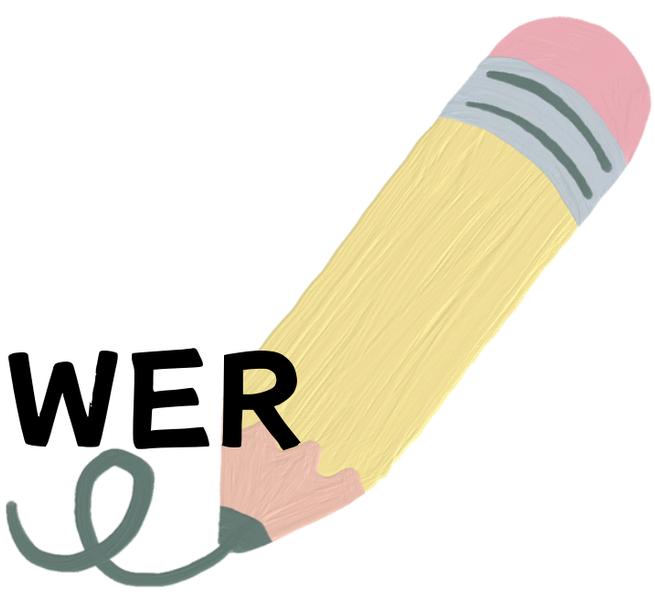
## QUESTION 3:

The program will ask for a 5-character input. Then, this 5-character output is displayed in reverse.

Assume input is abcde and desired output is edcba

# QR ANSWER





## SCAN THE QR FOR ANSWER

# EXERCISES

## QUESTION 4:

Write a program to print the area of a circle,
The radius of the circle is read from the user program.

# REFERENCES


- Dayang Norhayati Abang Jawawi &Rosbi Mamat. Pengenalan Mudah Pengaturcaraan C. Universiti Teknologi Malaysia, Johor.

- Nor Haizan Mohamed Radzi, Siti Zaiton Mohd Hashim, Faridah Shamsuri & Halizah Basiroh. Modul Pengajaran, Pengaturcaraan C. Universiti Teknologi Malaysia, Johor.

- Yashavant Kanetkar. Let Us C. Authentic guide to C programming language -16th Edition.

- Mike McGrath.C Programming in Easy Steps, version 6.3.0- 5th Edition.

# REFERENCES


- Exercise with online by W3resource.
https://www.w3resource.com/index.php

- Learning C Language eBook (PDF)
https://riptutorial.com/ebook/c

- C exercise- Practive Questions with solution
https://www.geeksforgeeks.org/c-exercises/

- Learn to code by doing. Try hands-on C programming with programmiz
https://www.programiz.com/c-programming/examples1