

Chapter 1. NUMBER SYSTEM

Page No. 9

I. Choose the correct answer

- b) 8
- a) Digital
- c) 16
- a) Most Significant Bit
- b) Octal

II. Answer the following questions:

- A bit is a small piece of data that is derived from the words "Binary digit."
- A number system denotes a group of characters which denote various numerical quantities.
- Binary, Octal, Decimal, Hexadecimal
- Sound waves, Telephone signals, Temperatures
- MSB - The left most bit in the binary number is called Most Significant Bit.
LSB - The right most bit in the binary number is called Least Significant Bit

III. Solve the following:

$$\begin{array}{r}
 1 \quad 1 \quad 1 \\
 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \\
 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \\
 (+) \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \\
 \hline
 (1011001)_2 + (110011)_2 = (100011100)_2
 \end{array}$$

$$\begin{array}{r}
 0 \quad 10 \\
 1 \quad 1 \quad 0 \quad 1 \\
 1 \quad 0 \quad 1 \quad 0 \\
 (+) \quad 0 \quad 0 \quad 1 \quad 1 \\
 \hline
 (1101)_2 - (1010)_2 = (0011)_2
 \end{array}$$

$$\begin{aligned}
 3. \quad (i) \quad (F4)_{16} &= (11110100)_2 \\
 (ii) \quad (CAB)_{16} &= (110010101011)_2
 \end{aligned}$$

$$4. \quad (i) \quad (32)_{10}$$

2	32	
2	16	0
2	8	0
2	4	0
2	2	0
	1	0

$$(32)_{10} = (100000)_2$$

$$(ii) \quad (65)_{10}$$

2	65	
2	32	1
2	16	0
2	8	0
2	4	0
2	2	0
	1	0

$$(65)_{10} = (1000001)_2$$

$$5. \quad (i) \quad \underbrace{(0001)}_1 \underbrace{(1010)}_A = (1A)_{16}$$

$$(ii) \quad (0111)_2 = (7)_{16}$$

IV. Answer the following:

- Binary number system
 - Octal number system
 - Decimal number system
 - Hexa decimal number system

Binary number system: In the binary number system, only the digits, 0s and 1s, are used.
Ex.: $(11011)_2$

Octal number system: The octal number system, which has the base 8, contains eight symbols(0-7).
Ex.: $(426)_8$

Decimal number system: The system that uses the decimal numbers or digits 0 to 9 is called the decimal number system. This system is said to have a base, or radix, of ten. Ex.: $(597)_{10}$

Hexa decimal number system: The sixteen symbols that are used in the hexadecimal system are 0, 1, 2, .. 9, A, B, C, D, E and F. Ex.: $(2A7)_{16}$

2. Convert the following :

1. (i)

2	516	
2	258	0
2	129	0
2	64	1
2	32	0
2	16	0
2	8	0
2	4	0
2	2	0
	1	0

$$(516)_{10} = (1004)_8$$

(ii)

8	516	
8	64	4
8	8	0
	1	0

$$(516)_{10} = (204)_{16}$$

(iii)

16	516	
16	32	4
16	2	0

$$(516)_{10} = (1000000100)_2$$

Chapter – 2: FUNCTIONS IN “C”

Page No. 22

I. Match the following:

stdio.h	-	This is a standard input/output header file. It contains all the library functions Regarding standard input/output.
conio.h	-	This is a console input/output header file
stdlib.h	-	This header file contains all the general library functions like malloc(), calloc(); exit(), etc...

string.h	-	It contains all string related library functions like get(), puts(), etc...
math.h	-	This header file contains all the math operations related functions like sqrt(), pow(), etc.....

II. Answer the following:

1. A function is a set of code that take inputs, do some specific computation and produces output.
2. • Functions help us in reducing code redundancy, which means repeating a set of code repeatedly.
• Functions make code modular
• Functions provide abstraction
3. By using function prototype we declare a function in “C”

Function Prototype

Header • return data type function name
(parameter)

Body • statements return 0;

III. Answer in detail:

1. Refer Q1, Advantages of Functions in C, Q3
2. Refer the book page no.20 and 21

IV. Lab time:

2.

```
#include <stdio.h>
```

```
int main() {
```

```
    int number;
```

```
    printf("Enter a number: ");
```

```
    scanf("%d", &number);
```

```
    if (number % 2 == 0) {
```

```
        printf("%d is an even number.\n", number);
```

```
    } else {
```

```
        printf("%d is an odd number.\n", number);
```

```
    }
```

```
    return 0;
```

```
}
```

Chapter – 3: VARIABLES AND CONSTANTS IN “C”

Page – 29.

I. Choose the right answer:

1. b) int count;
2. a) True
3. a) Const
4. b) Static

II. Answer the following:

1. In “C”, a variable has a name and contains a value.
2. Constants are values that are fixed throughout the program.
3.
 1. Local
 2. Global
 3. Static
 4. External
 5. Automatic
4. External variable can be shared across multiple “C” program files, this variable is declared using the extern keyword.

III. Answer in detail:

1. Refer Book Page No. 25, 26

2.

```
#include <stdio.h>
{
    float a, b, avg;
    printf (" Enter first number ");
    scanf ("%f", &a);
    printf (" Enter second number ");
    scanf ("%f", &b);
    avg=(a+b)/2;
    printf (" Average of two numbers = %f", avg);
    return 0;
}
```

3.

```
#include <stdio.h>
int main()
{
    int a, b, c;
    printf (" Enter first number");
```

```
scanf ("%d", &a);
printf ("Enter second number");
scanf ("%d", &b);
c = a-b;
printf ("Difference of two numbers is = %d", c);
return 0;
}
```

IV. Project Time:

1.

```
#include <stdio.h>
```

```
int main() {
    int num1, num2;
```

```
    printf ("Enter the first number: ");
    scanf ("%d", &num1);
```

```
    printf ("Enter the second number: ");
    scanf ("%d", &num2);
```

```
    // Adding and subtracting 20 and 30 from
    the numbers
```

```
    int added_num1 = num1 + 20;
    int added_num2 = num2 + 20;
    int subtracted_num1 = num1 - 30;
    int subtracted_num2 = num2 - 30;
```

```
    printf ("After adding 20, the first number
    becomes: %d\n", added_num1);
    printf ("After adding 20, the second number
    becomes: %d\n", added_num2);
    printf ("After subtracting 30, the first number
    becomes: %d\n", subtracted_num1);
    printf ("After subtracting 30, the second
    number becomes: %d\n", subtracted_num2);
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
```

```
int main() {
    int num1, num2;
```

```
    printf ("Enter the first number: ");
    scanf ("%d", &num1);
```

```
    printf ("Enter the second number: ");
    scanf ("%d", &num2);
```

```

int sum = num1 + num2;
int difference = num1 - num2;
int product = num1 * num2;
int quotient = num1 / num2;
int remainder = num1 % num2;

printf("Sum: %d\n", sum);
printf("Difference: %d\n", difference);
printf("Product: %d\n", product);
printf("Quotient: %d\n", quotient);
printf("Remainder: %d\n", remainder);

return 0;
}

```

Chapter – 4: CONTROL STATEMENTS AND LOOPS IN C

Page No. 40-42

I Fill in the blanks:

1. Conditional or decision-making
2. Root directory
3. default
4. break
5. loop
6. Rename

II. True / False

1. True
2. True
3. True

III. Choose:

1. c) executes continuously
2. b) for loop
3. b) do while

IV. Project Time:

```

1.
#include <stdio.h>

int main() {
    int number;

```

```

printf("Enter a number: ");
scanf("%d", &number);
if (number % 3 == 0) {
    printf("%d is a multiple of 3.\n", number);
} else {
    printf("%d is not a multiple of 3.\n", number);
}

return 0;
}

2.
#include <stdio.h>

int main() {
    int a,b,c;

    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    if (a >= b && a >= c {
        printf("%d is the greatest number.\n", a);
    } else if (b >= a && b >= c) {
        printf("%d is the greatest number.\n",b);
    } else {
        printf("%d is the greatest number.\n", c);
    }
    return 0;

3.
Fibonacci Series.
# include <stdio.h>
int main()
{
    int i, n;
    int f1=0, f2=1;
    int f3=f1+f2;
    printf ( " Enter the number of terms ");
    scanf("%d", &n);

    printf(" Fibonacci Series %d %d ", f1, f2)
    for (i=3; i<=n, ++i)
    {
        printf("%d", f3);
        f1=f2;
        f2=f3;
        f3=f1+f2;
    }
    return 0;
}

```

4.

```
//for loop

#include <stdio.h>

int main() {
    int i;

    printf("Using for loop:\n");
    for (i = 0; i < 10; i++) {
        printf("Computer\n");
    }

    return 0;
}

//while loop
#include <stdio.h>

int main() {
    int i = 0;

    printf("Using while loop:\n");
    while (i < 10) {
        printf("Computer \n");
        i++;
    }

    return 0;
}

//do while loop

#include <stdio.h>

int main() {
    int i = 0;

    printf("Using do-while loop:\n");
    do {
        printf("Computer \n");
        i++;
    } while (i < 10);

    return 0;
}
```

Chapter – 5: ARRAYS IN “C”

Page No. 48-49

I. Fill in the blanks:

1. Array
2. index position

II. True or False

1. True
2. False

III. Choose the correct answer:

1. b) 0

IV. Project Time:

```
1.
#include <stdio.h>

// Function to search for a value in an array
int searchArray(int *arr, int size, int value) {
    int i;

    for (i = 0; i < size; i++) {
        if (arr[i] == value) {
            return i; // Return the index if value
is found
        }
    }

    return -1; // Return -1 if value is not found
}

int main() {
    int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int size = sizeof(arr) / sizeof(arr[0]);
    int searchValue;
    int result;

    printf("Enter the value to search: ");
    scanf("%d", &searchValue);

    // Call the searchArray function to find the val-
ue in the array
    result = searchArray(arr, size, searchValue);

    if (result != -1) {
```

```

        printf("Value %d found at index %d.\n",
            searchValue, result);
    } else {
        printf("Value %d not found in the array.\n",
            searchValue);
    }

    return 0;
}

```

2.

```

#include <stdio.h>
void bubbleSort(int arr[], int size) {
    int i, j, temp;

    for (i = 0; i < size - 1; i++) {
        for (j = 0; j < size - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

int main() {
    int arr[] = {5, 2, 8, 1, 6};
    int size = sizeof(arr) / sizeof(arr[0]);
    int i;

    printf("Original array: ");
    for (i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }

    bubbleSort(arr, size);

    printf("\nSorted array in ascending order: ");
    for (i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}

```

3.

```

#include <stdio.h>

```

```

int main() {
    int numGrades, i, numStudents, totalStrength = 0;

    printf("Enter the number of grades or classes: ");
    scanf("%d", &numGrades);

    for (i = 1; i <= numGrades; i++) {
        printf("Enter the number of students in grade or class %d: ", i);
        scanf("%d", &numStudents);
        totalStrength += numStudents;
    }

    printf("Total strength of the school: %d\n", totalStrength);

    return 0;
}

```

Chapter – 6: AN INTRODUCTION TO HTML AND FORMATTING TAGS EXISTING

Page No. 65-67

I. Choose the correct answer:

1. b) Hyper Text Markup Language
2. d) HTML
3. d) both a and b
4. a) Tim Berners Lee
5. b) <>
6. b) 6
7. a) left
8. a)

9. c)

10. c) No shade

II. Answer the following:

1. HTML stands for Hyper Text Markup Language. Hypertext is simply a piece of text that contains links to other texts. Markup language is a way of writing style and layout information within a text document.

2. • HTML Editor
• Web Browser
• Web Server
3. HTML editor is a software application required to create web pages. Some editors are:
 - WYSIWYG Editor
 - Text Editor
4. WYSIWYG stands for What You See Is What You Get.
5. • Internet Explorer
• Mosaic
• Netscape Navigator
• Net Cruiser
• Mac web/Win Web
• Mozilla Firefox
• Opera
6. A tag is an element that instructs the web browser what to display and how to display it.
7. A web server is a software that uses HTTP (Hyper Text Transfer Protocol) to serve the files that form web pages to users.
8. A tag that has both a start tag and an end tag is called a container tag.
Ex. <HTML>.....</HTML>
 <BODY>.....</BODY>
9. Tags that have only a start tag and do not contain text or any other tag elements are called empty tags.

, <HR>.....
10. • Head Section
 • Body Section
11. Heading tags used to provide a heading to a web page.
12. <HTML>
 <HEAD>
 <TITLE> Paragraph alignment </TITLE>
 </HEAD>
 <BODY>
 <Palign = "left"> unity is strength </p>
 <Palign = "center"> unity is strength </P>
 <Palign = "right"> unity is strength </P>
 </BODY>
 </HTML>
13. Center tag displays the text or an image in the center of the web page.

14. Size attribute is used to increase or decrease the font size.
15. Superscript tag is used to display the text in the superscript format.
Subscript tag is used to display the text in the subscript format.

III. Answer in detail:

1. An HTML document has two sections,
 - Head Section
 - Body Section

Head Section: The head section contains information about the document.

Body Section: The body section defines the document's body. This section may contain several other tags.

Head Section:

```
<HTML>
<HEAD>
<TITLE>.....</TITLE>
</HEAD>
```

Body Section:

```
<BODY>
</BODY>
</HTML>
```

<TITLE> - This is used to provide the title to as HTML document

</TITLE> - This tag is used to end a title.

<HEAD> - This tag is used to provide a heading to a web page

</HEAD> - This tag is used to end the heading

<BODY> - This tag hold the body of the page

</BODY> This tag is used to end the body

2. To create, save, and view an HTML document, you can follow these steps:
 1. **Create an HTML file:** Open a text editor (e.g., Notepad) and create a new file. Save it with a .html extension, such as example.html.
 2. **Write HTML code:** Inside the HTML file, write your desired HTML code. For example, you can create a simple HTML structure with a heading and a paragraph:
 3. **Save the HTML file:** Save the file after writing the HTML code.

4. **View the HTML document:** Open the saved HTML file in a web browser (e.g., Google Chrome, Mozilla Firefox). You can either double-click the file or right-click and choose "Open with" to select a browser.
3. an HTML code to display your address.


```
<HTML>
<HEAD>
<TITLE> my address </TITLE>
</HEAD>
<BODY>
XXXX
Siddhu Matric. Hr. Sec. School
5, Vijayalakshmi Theatre Road,
Viswanathapuram,
Madurai-14
</BODY>
</HTML>
```
4. To reduce the size of the text from the normal size, we use <Small> tag. To display a text bigger than the normal size, we use <Big> tag.
5.
 1. ** and :** These tags are used to make text bold. They are typically used to emphasize important or strong content.
 2. **<i> and :** These tags are used to italicize text. They are commonly used to indicate emphasis or to differentiate a particular section of text.
 3. **<u>:** This tag is used to underline text. It is often used to highlight or indicate links or important information.
 4. **<s> and <strike>:** These tags are used to strikethrough text. They are often used to indicate deleted or outdated content.
 5. **<sub>:** This tag is used to render subscript text. It is commonly used for mathematical or chemical formulas where smaller text is placed below the baseline.
 6. **<sup>:** This tag is used to render superscript text. It is commonly used for mathematical equations or to indicate footnotes.

Chapter – 7: HTML – Level II

Page No. 75

I. Choose the correct answer:

1. d) all of these
2. b) double quotes
3. c) 3
4. c) BG Sound
5. a) 3

II. Answer the following:

1. We can change the font of the text in HTML using the face attribute of tag.
2. List out different lists in HTML.
 - Ordered List
 - Unordered List
 - Definition List
3. Moving or scrolling text can be placed in a web page by using <marquee> tag.
4. To change the background color of a web page, by color attribute is used in the body tag.
5. The <BGSOUND> tag is used to add sound for HTML.

III. Answer in detail:

1. Refer book page no.71
2. Refer book page no.73
3. Refer book page no.73
4. Refer book page no.74

Chapter 8. MORE ON EV3

Page No. 80

I. Fill in the blanks:

1. Color
2. 7
3. loop
4. Orange



II. Answer the following:

1. The color sensor is used to make the EV3 follow a line, find a color, move until reaching a line and other such operations.
2. To repeat a set of blocks is called a loop.
3. black, brown, blue, green, yellow, red, white

III. Answer in detail:

1. Refer the Book Page No. 76
2. Refer the Book Page No. 79 & 80

IV. Project time:

Line Follower Program with Touch Sensor Stop

Import the necessary modules

```
from ev3dev2.motor import OUTPUT_A, OUTPUT_B, MoveTank
```

```
from ev3dev2.sensor import INPUT_1, INPUT_2, INPUT_3
```

```
from ev3dev2.sensor.lego import ColorSensor, TouchSensor
```

Initialize the sensors and motors

```
tank = MoveTank(OUTPUT_A, OUTPUT_B)
```

```
color_sensor = ColorSensor(INPUT_1)
```

```
touch_sensor = TouchSensor(INPUT_3)
```

Set the desired speed and adjust the following values as needed

```
speed = 20
```

```
threshold = 30
```

Main program loop

```
while not touch_sensor.is_pressed:
```

```
    # Read the reflected light intensity from the color sensor
```

```
    intensity = color_sensor.reflected_light_intensity
```

```
    # Adjust the motor speeds based on the intensity of the reflected light
```

```
    if intensity < threshold: # Line is to the right, turn left
```

```
        tank.on(left_speed=speed, right_speed=0)
```

```
    else: # Line is to the left, turn right
```

```
        tank.on(left_speed=0, right_speed=speed)
```

```
# Stop the robot when the touch sensor is pressed
tank.off()
```

Chapters 9. IoT ON MICRO:BIT

Page No. 84

I. Fill in the blanks

1. 25
2. BBC
3. magnetometer
4. loop
5. 255

II. Answer the following

1. sensor is a hardware component that detects and measures physical quantities or environmental conditions.
2. Accelerometer
Magnetometer
Temperature Sensor
Light Sensor
3. Refer the book pg no 81
4. Refer the book pg no 81
5. Measures the sound levels in numbers between 0 and 255.

III. Answer in detail

1.

```
from microbit import *

while True:
    sound_value = microphone.sound_level()

    if sound_value > 200:
        display.show(Image.ASLEEP)
        music.play(music.BA_DING)
        display.clear()
    else:
        display.show(Image.HAPPY)
```
2. Refer the book page no. 83.