Chapter 7: Input and output

I. getchar() / getc()

1. Write a program using getchar() or getc() to receive a character and print the character and its integer value.

2. Write a program,which uses getchar() or getc(), receives 4 characters, print them and their integer value out.

II. putchar()

3. Using the same idea from example at pages 86-87, create a program that prints the word "UwU"

III. char array

4. write a program using fgets() to receive your date of birth, then print it out.

Chapter 8: Decision Making

1. Write a program to check whether a given year is a leap year or not.
2. Write a program to check whether a number is a multiple of 7s or not.

3. Write a program to check whether a number is negative, positive. Putting a different output for number 0.

4. Write a program to calculate the root of a quadratic equation

5. Write a 'guessing number' program: Every time this program runs, a random number from 0 - 10 is generated, name it the solution. The user input the same number as 'the solution' to stopped the program.

Chapter 9 :Loops

1. Write a program using 'for' loop to print number from 1 to 68
2. Write a program to display the multiplication table of a given integer

Test:
Input the number : 6
Expected output:

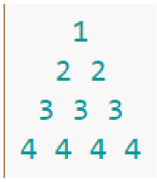
```
6 X 1 = 6
6 X 2 = 12
6 X 3 = 18
.
.
.
6 X 10 = 60
```

3. Write a program to display the n terms of odd number and their sum

Test Data
Input number of terms : 5
Expected Output :

```
The odd numbers are :1 3 5 7 9
The Sum of odd Natural Number upto 5 terms : 25
```

4. Write a program to make a pyramid with a number which repears the number in the same row



```
   1
  2 2
 3 3 3
4 4 4 4
```

Chapter 10: Function

1. Write a program using *void* function to print string to the screen with your signature

- That function should look like:

```
myPrintFunction("Hello  World")
myPrintFunction("Nice  to meet you")
```

- Output:

```
Hello World - From Long
Nice to meet you - From Long
```

2. Write a program using function to check if a given number is even or odd

Test Data :

Input any number : 69

Expected Output :

```
 The entered number is odd.
```

3. Write a program using function to swap value between a pair of variables

4. Write a program using function to calculate the sum of all the numbers from 1 to n

5. Write a program to check whether two given strings are an anagram

- An **anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

Test Data :

Input the first String : spare

Input the second String : pears

Expected Output :

spare and pears are Anagram.

Chapter 11:

1. What is the function of these operators ?

++
--
+=
-=
*=
%=
%

2. What is the different between these two equations?

a-=2021	a=-2021

3. What do these functions do?

sqrt()
pow()
abs()
ceil()
floor()

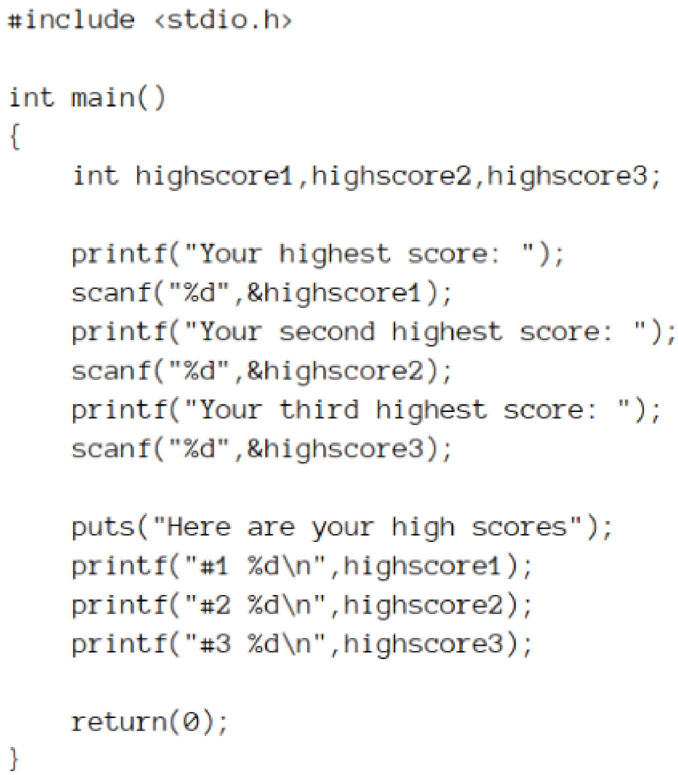
4. Write a program which uses your own function to calculate x raised to the power of n (x^n)

5. Write a program to reverse the digits of a given integer

Example : 123 -> 321

Chapter 12: Array

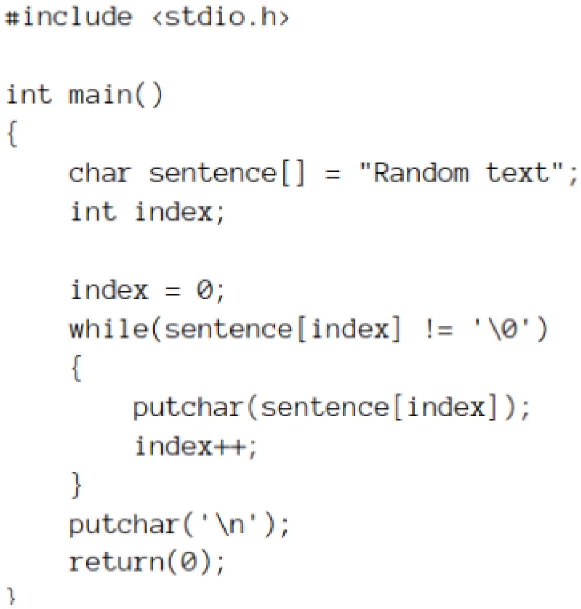
1. From the example in page 174 of the Book, explain why array is useful in some situation



```
#include <stdio.h>

int main()
{
    int highscore1,highscore2,highscore3;

    printf("Your highest score: ");
    scanf("%d",&highscore1);
    printf("Your second highest score: ");
    scanf("%d",&highscore2);
    printf("Your third highest score: ");
    scanf("%d",&highscore3);

    puts("Here are your high scores");
    printf("#1 %d\n",highscore1);
    printf("#2 %d\n",highscore2);
    printf("#3 %d\n",highscore3);

    return(0);
}
```

? - Can you improve this printf() method by using array and loops ? Write a program for it

2. Take a look at this code



```
#include <stdio.h>

int main()
{
    char sentence[] = "Random text";
    int index;

    index = 0;
    while(sentence[index] != '\0')
    {
        putchar(sentence[index]);
        index++;
    }
    putchar('\n');
    return(0);
}
```

? Explain the meaning of the line: while(sentence[index] != '\0'), why is it so important to determine the '\0' character of this type of array?

3. Write a program using fgets() that receives your name, day of birth, and your hometown then print it out to the screen.

4. Write a program to calculate the sum of all array elements.

5. Write a program in C for a two dimensional array of size nxn and print it

Chapter 13: CTYPE

1. Fill in the tables below

Function	**Returns True When *a* is**
isalnum(a)	
isalpha(a)	
iscntrl(a)	
isdigit(a)	
isgraph(a)	
islower(a)	
isprint(a)	
ispunct(a)	
isspace(a)	
isupper(a)	
isxdigit(a)	

- What would tolower(a) and toupper(a) return ?

2. Create a program that converts all the upper characters of a given string to lower, all the lower characters to upper

For example:
Input : hEllo World
Output: HeLLO wORLD

3. Create a program that counts the number of spaces in a given sentence

For example:
Input: Hello Hello Hello
Output: 2

4. Create a program that concatenate two strings together

Chapter 14: Struct

1. Read the code below

```
struct record
{
char name[32];
int age;
float debt;
} bill, mary, dan, susie;
printf("Victim: %s\n",bill.name);
```

? What is the output of this code ? Explain ? Can you fix this problem ?

2. Assign information of 4 records in Ex1

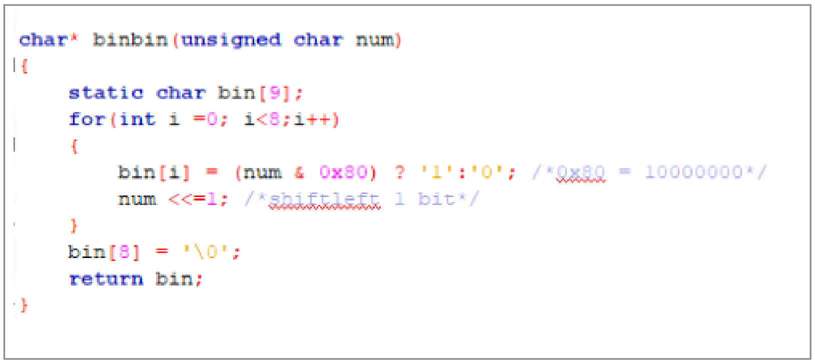
	bill	mary	dan	susie
name	Bill	Mary	Dan	Susie
age	20	21	22	23
debt	500	700	2600	1800

3. Create an array that stores all 4 records in Ex2

Chapter 16:

Chapter 17: Binary Mania

Read the function below



```
char* binbin(unsigned char num)
{
    static char bin[9];
    for(int i =0; i<8;i++)
    {
        bin[i] = (num & 0x80) ? '1':'0'; /*0x80 = 10000000*/
        num <<=1; /*shiftleft 1 bit*/
    }
    bin[8] = '\0';
    return bin;
}
```

1 - Based on the binbin() function, write a binbin() function for integer number.
unsigned char -> 8 bits then int -> 4 bytes -> 32 bits

2 - Based on the binbin() function, write a function to check whether an integer number is a power of 2 or not.
Using &

3 - Base on the OR bitwise, NOT bitwise and the source code in the textbook, write a function to convert lowercase letter to upper case letter.
Using |, ~

4: Write a function to print an integer number to hexadecimal number. Use the stdin to get the input number

5: Write a function Multiplier(int number, int power), which will return the result of the multiplication between the input number and a power of 2

Chapter 18: Introduction to Pointer

1: Define 3 variables, 1 int, 1 double and 1 char, then print the size of each variable.
// Using sizeof() function

2: Use strlen() and sizeof() to get the length of a string. Explain the result, by comment.

3: Define a variable a = 10. Change the value in a to 100 by 2 ways.

4: Create an array of integer numbers. Print the address of its elements in a row.
//Defind a pointer pointing to array[3] and a pointer pointing to array[4]
//Increase the pointer pointing array[3] by 1. Comment.

5: Define an array of integer numbers. Write a code to get the length of the array.

Chapter 19: Deep into Pointer Land

1: Define a variable a = 10. Write 2 VOID functions to increase value of a by 1. Print a once again.
//2 Ways: Using variable - Using pointer

2: Define an array of integer numbers. Define a pointer *p pointing to that array. Print the elements of the array using *p

3: Define an array of pointers, where each element is a string with length at least 2. Print the second character of each string

4: Define a void function named ExchangeUsd2Vnd(double* money). This function will take a double type pointer as argument. The
//function will convert the money in Usd to Vnd. Note: Vnd = Usd*23

Chapter 20: MemoryChunkand LinkedList

1: Define a pointer name p. Now use p to store number from 0 to 9. Print the numbers. Hint: Use malloc() to allocate a buffer
//for p to point to

2: Exercise 20-2 in Textbook.

3: From EX1, realloc the buffer so that p can hold up to 20 numbers.

Chapter 21: Its about time

1: Exercise 21-4: Write code that passes the time_t value 946684800 to the ctime() function. Output the string that’s returned.

2:Exercise 21-7: Write code that outputs the current time in the format hour:minute:second. Ensure that the minute-and-second output is two digits

3:Exercise 21-8: Fix your solution from Exercise 21-7 so that the output is in 12-hour format with an A.M. or P.M. suffix based on the time of day.
4: Modify the code in listing 21-3 so that the code print the correct year.

Chapter 22: Permanent Storage

1: Write a line "Hello World" to file hello.txt.

2: Read data from file hello.txt and print it to the screen.

3: Based on READ and WRITE functions, write a code for copying file hello.txt to hello_copied.txt.

4: Write a code to check whether 2 files are the similar or not.

5: File dumper: Write a code to print the file hello.txt into a matrix of width 5, in hexadecimal form.

Chapter 23: File Handling

1.Write a program in C to create and store information in a text file
2. Write a program in C to read an existing file.
3. Write a program in C to write multiple lines in a text file.
4. Write a program in C to find the content of the file and number of lines in a Text File.