Assignment 1

WAP to check whether a given number is odd or even and its signedness

Description:

- The user entered number should identified whether its a odd or a even number. Mention its sign too.

Pr-requisites:-
- Loops
- Arithmetic Operators
- Nested if-else construct

Objective: -
- To understand the concept of
  - Nested if-else

Inputs: -

Integer 'N'. Where 'N' < 2^{20}

Sample execution: -

Test Case 1:

user@emertxe] ./even_odd
Enter the value of 'n' : -4
-4 is -ve even number
Do you want to continue(y/n): y
Enter the value of 'n' : 4
4 is +ve even number
Do you want to continue(y/n): y
Enter the value of 'n' : 2000000
Number out of range
Do you want to continue(y/n): n
user@emertxe]

Assignment 2

WAP to generate fibbonacci numbers <= 'n'

Description:

- In mathematics, the Fibonacci numbers or Fibonacci sequence are the numbers in the following integer sequence 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 . . . OR 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 . . .
- By definition, the first two numbers in the Fibonacci sequence are either 1 and 1, or 0 and 1, depending on the chosen starting point of the sequence, and each subsequent number is the sum of the previous two

Pr-requisites:-
- Loops
- Arthmetic operators

Objective: -

- To understand the concept of
  - Continuos looping.
  - If-else constructs

Inputs: -

A integer say 'N'

Sample execution: -

Test Case 1: Positive Numbers
user@emertxe] ./fibbonacci_series
Enter a number: 8
0, 1, 1, 2, 3, 5, 8

Test Case 2: Positive Numbers
user@emertxe] ./fibbonacci_series
Enter a number: 10
0, 1, 1, 2, 3, 5, 8

Test Case 3: Negative Number
user@emertxe] ./fibbonacci_series
Enter a number: -21
0, 1, -1, 2, -3, 5, -8, 13, -21

Test Case 4: Negative Number
user@emertxe] ./fibbonacci_series
Enter a number: -13
0, 1, -1, 2, -3, 5, -8, 13

Assignment 3

WAP to check whether a given number is perfect or not

Description:

- In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum).
- Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).

Example:

- The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6
- The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128
-

Pr-requisites:-

- Loops
- Arthmetic operators

Objective: -

- To understand the concept of
  - If-else constructs

Inputs: -

A positive integer say 'N'. Where N < 2^{20}

Sample execution: -

Test Case 1: Positive Numbers
user@emertxe] ./perfect_number
Enter a number: 6
Yes, entered number is perfect number
Test Case 2: Positive Numbers
user@emertxe] ./perfect_number
Enter a number: 10
No, entered number is not a perfect number
Test Case 3: Negative Number
user@emertxe] ./perfect_number
Enter a number: -1
Invalid Input
Test Case 4: N > 2^{20}
user@emertxe] ./perfect_number
Enter a number: 2000000
Number out of range

Assignment 4

To Print all ascii characters

Description:

- Print the Decimal numbers from 0 to 127 in Octal, Hexadecimal and in ASCII representation.
- Display as 'Non Printable' for non-printable characters

Pr-requisites:-

- Loops

Objective: -

- To understand the concept of
  - Loops and ASCII characters

Inputs: -

None

Sample execution: -

Test Case 1:
user@emertxe] ./ascii

Dec	Oct	Hex	Ascii
0	000	00	Non Printable
1	001	01	Non Printable

Assignment 5

To Print sizes of all basic data types of C

Description:

- Print the size of basic data types – int, char, float, double, void
- Print the size of the above data types while using different qualifiers – short, long, unsigned etc.

Pr-requisites:-
- Sizeof operators

Objective: -
- To understand the concept of
  - Basic C datatypes and its size

Inputs: -

None

Sample execution: -

Test Case 1:

user@emertxe] ./sizeof

```
Size of int           : 4 bytes
Size of char          : 1 byte
Size of float         : 4 bytes
Size of double        : 8 bytes
Size of unsigned int  : 4 bytes
Size of long int      : 8 bytes
----
----
```

Assignment 6

Given a number n, WAP to print all primes smaller than or equal to n. Use Sieve of Eratosthenes method

Description:

- In mathematics, the sieve of Eratosthenes, one of a number of prime number sieves, is a simple, ancient algorithm for finding all prime numbers up to any given limit. It does so by iteratively marking as composite (i.e., not prime) the multiples of each prime, starting with the multiples of 2.
- The sieve of Eratosthenes is one of the most efficient ways to find all of the smaller primes. It is named after Eratosthenes of Cyrene, a Greek mathematician.

Pr-requisites:-
- Loops
- Arithmetic Operators
- Arrays

Objective: -

- To understand the concept of
  - If-else constructs
  - Arrays

Inputs: -

Integer 'N'. Where 'N' < 2^{20}

Sample execution: -

Test Case 1: A positive number

user@emertxe] ./prime_series
Enter the value of 'n' : 20
The primes less than or equal to 20 are : 2, 3, 5, 7, 11, 13, 17, 19

Test Case 2: A negative number

user@emertxe] ./prime_series
Enter the value of 'n' : -20
Invalid Input

Test Case 3: Number greater than 2^{20}

user@emertxe] ./prime_series
Enter the value of 'n' : 2000000
Number out of range

Assignment 7

WAP to read 3 numbers a, r, n. Generate AP, GP, HP

Description:

- AP
  - In mathematics, an arithmetic progression (AP) or arithmetic sequence is a sequence of numbers such that the difference between the consecutive terms is constant.
  - For instance, the sequence 5, 7, 9, 11, 13, 15 ... is an arithmetic progression with common difference of 2.
- GP
  - In mathematics, a geometric progression, also known as a geometric sequence, is a sequence of numbers where each term after the first is found by multiplying the previous one by a fixed, non-zero number called the common ratio.
  - For example, the sequence 2, 6, 18, 54, ... is a geometric progression with common ratio 3. Similarly 10, 5, 2.5, 1.25, ... is a geometric sequence with common ratio 1/2.
- HP
  - In mathematics, a harmonic progression (or harmonic sequence) is a progression formed by taking the reciprocals of an arithmetic progression.

Pr-requisites:-

- Loops
- Arthmetic operators
- Data Types

Objective: -

- To understand the concept of
  - Continuos looping.
  - If-else constructs
  - Type Casting

Inputs: -

Positive integers say 'A', 'R' and 'N'
where:
A = First number
R = Common difference(AP & HP), Common ratio(GP)
N = number of terms
A, R and N should be < 2^{10}

Sample execution: -

Test Case 1: Positive Inputs
user@emertxe] ./progressions
Enter the First Number 'A': 2
Enter the Common Difference / Ratio 'R': 3
Enter the number of terms 'N': 5
AP = 2, 5, 8, 11, 14
GP = 2, 6, 18, 54, 162
HP = 0.5, 0.2, 0.125, 0.0909091, 0.0714285

Test Case 2: Negative Input(s)
user@emertxe] ./progressions
Enter the First Number 'A': -2
Enter the Common Difference / Ratio 'R': 3
Enter the number of terms 'N': 5
Invaild Input(s)

Test Case 3: Any one value greater than 2^{10}
user@emertxe] ./progressions
Enter the First Number 'A': 2000
Enter the Common Difference / Ratio 'R': 3
Enter the number of terms 'N': 5
Input(s) out of range

Assignment 8

WAP to print “Hello” in X format

Description:

- When the program is run, the provided string should be printed on the terminal in X formation

Pr-requisites:-

- Loops
- Arthmetic operators

Objective: -

- To understand the concept of
  - Continuos looping.
  - If-else constructs

Inputs: -

A integer say 'N'. Where N < 2^5

Sample execution: -

Test Case 1:

```
user@emertxe] ./x_pattern
Enter a number lines: 5
Hello       Hello
  Hello    Hello
    HelloHello
  Hello    Hello
Hello       Hello
```

Test Case 2:

```
user@emertxe] ./x_pattern
Enter a number lines: 32
Number out of range
```

Test Case 3:

```
user@emertxe] ./x_pattern
Enter a number lines: 6
Hello       Hello
  Hello    Hello
    HelloHello
    HelloHello
  Hello    Hello
Hello       Hello
```

Assignment 9

WAP to find the median of two unsorted arrays

Description:

- In probability theory and statistics, a median is described as the number separating the higher half of a sample, a population, or a probability distribution, from the lower half. The median of a finite list of numbers can be found by arranging all the numbers from lowest value to highest value and picking the middle one.

- Example
  - For getting the median of input array { 12, 11, 15, 10, 20 }, first sort the array. We get { 10, 11, 12, 15, 20 } after sorting. Median is the middle element of the sorted array which is 12.

Pr-requisites:-
- Loops
- Arrays

Objective: -
- To understand the concept of
  - One-dimensional Arrays

Inputs: -

2 Integer Array of 'N' elements. Where 'N' < 10

Sample execution: -

Test Case 1: Equal sized arrays
user@emertxe] ./median
Enter the 'n' value for Array A: 5
Enter the elements one by one for Array A: 3 2 8 5 4
After sorting : 2 3 4 5 8
Median of array1 : 4
Enter the 'n' value for Array B: 5
Enter the elements one by one for Array A: 12 3 7 8 5
After sorting : 3 7 5 8 12
Median of array2 : 5
Median of both arrays : 4.5 (4 + 5 = 9; 9 / 2 = 4.5)

Test Case 2: Unequal sized arrays
user@emertxe] ./median
Enter the 'n' value for Array A: 5
Enter the elements one by one for Array A: 3 2 8 5 4
After sorting : 2 3 4 5 8
Median of array1 : 4
Enter the 'n' value for Array B: 4
Enter the elements one by one for Array A: 12 13 7 5
After sorting : 5 7 12 13
Median of array1 : 9.5 (7 + 12 = 19; 19 / 2 = 9.5)

Median of both arrays : 6.75 (4 + 9.5 = 13.5; 13.5 / 2 = 6.75)

Test Case 3: N greater than 10
user@emertxe] ./median
Enter the 'n' value for Array A: 20
Max array elements exceeded

Assignment 10

Given the number from 1 to 365, WAP to find which day of the year

Description:

- Suppose, in a week let us assume first day is 'Sunday', then second day will be 'Monday' and so on. If first day is 'Monday' then the second day will be 'Tuesday' and so on.

Pr-requisites:-
- Loops
- Arthmetic operators
- Switch Case

Objective: -
- To understand the concept of
  - Switch Case

Inputs: -
Positive integer say 'N'. Where 1 <= 'N' <= 365.
Option to set the first day.

Sample execution: -
Test Case 1:
user@emertxe] ./day_of_the_year
Enter the value of 'n' : 9
Choose First Day :
1. Sunday
2. Monday
3. Tuesday
4. Wednesday
5. Thursday
6. Friday
7. Saturday
Enter the option to set the first day : 2
The 9th day is Tuesday
Do you want to continue(y/n): y
Enter the value of 'n' : 9
Enter the option to set the first day : 3
The 9th day is Wednesday
Enter the value of 'n' : 9
Enter the option to set the first day : 8
Invalid input

Do you want to continue(y/n): y
Enter the value of 'n' : 0
Invalid Input
Do you want to continue(y/n): y
Enter the value of 'n' : 366
Invalid Input

Assignment 11

To check a given number is even or odd using bitwise operators

Description:

- Read number n from user.
- Check the zeroth bit(on LSB end) of the number.
  - if last bit is 1, number is odd.
  - if last bit is zero, number is even.
- Prompt for continue option.

Pr-requisites:-
- If Else
- Bitwise Operators

Objective: -
- To understand the concept of
  - Bitwise Operators

Inputs: -
Integer N

Sample execution: -
Test Case 1:
user@emertxe] ./even_odd
Enter the value of 'N' : 4
4 is even number
Test Case 2:
user@emertxe] ./even_odd
Enter the value of 'N' : 5
5 is odd number

Assignment 12

To print bits of signed and unsigned types for given number and check for 2’s complement

Description:

- Read number n from user.
- Print the 32 bit binary representation of n and -n.
- And compare both.
- Prompt for continue option.

Pr-requisites:-

- Bitwise Operators

Objective: -

- To understand the concept of
  - 2's Compliment and Bitwise Operators

Inputs: -

Integer N

Sample execution: -

Test Case 1:

user@emertxe] ./2s_comp
Enter the number : 12
+12 -> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
-12 -> 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0

Assignment 13

Implement your own c-type library (any four).

Description:

- c-type library functions check whether c, which must have the value of an unsigned char or EOF, falls into a certain character class according to the current locale.
  - isalnum() - checks for an alphanumeric character; it is equivalent to (isalpha(c) || isdigit(c)).
  - isalpha() - checks for an alphabetic character; in the standard "C" locale, it is equivalent to (isupper(c) || islower(c)). In some locales, there may be additional characters for which isalpha() is true—letters which are neither upper case nor lower case.
  - isascii() - checks whether c is a 7-bit unsigned char value that fits into the ASCII character set.
  - isblank() - checks for a blank character; that is, a space or a tab.

Pr-requisites: -

- Loops
- Functions

Objective: -

- To understand the concept of
  - Functions

Inputs: -

An ASCII character

Outputs: -

0 or non-zero value based on condition success or failure

Sample execution: -

Test Case 1:

user@emertxe] ./c_type_lib
Enter the character: a
Select any option:
1 - isalpha
2 - isalnum
3 - isascii
4 - isblank
Enter you choice: 2
The character 'a' is an alnum character.

Test Case 2:
user@emertxe] ./c_type_lib
Enter the character: ?
Select any option:
1 - isalpha
2 - isalnum
3 - isascii
4 - isblank
Enter you choice: 2
The character '?' is not an alnum character.

Assignment 14

Implement the below mentioned bitwise functions

```
int get_nbits(int num, int n);
int replace_nbits(int num, int n, int val);
int get_nbits_from_pos(int num, int n, int pos);
int replace_nbits_from_pos(int num, int n, int pos, int val);
int toggle_bits_from_pos(int num, int n, int pos);
int print_bits(unsigned int num, int n);
```

Description:

- **get_nbits**

  If num is 10 and n is 2,
  10 -> 0 0 0 0 1 0 1 0
  Take 2 bits from LSB end of 10 (1 0) and return the corresponding decimal of that.
  So get_nbits(10, 2) function should return 2

- **replace_nbits**
  ---
  a. Read number num from user.
  b. Read number n from user.
  c. Read number val from user
  d. Fetch n number of bits from LSB end of val and replace in the last n bits of num.
  e. Return new value of num.
  If num is 10 and n is 3 and val is 12
  10 -> 0 0 0 0 1 0 1 0

---
12 -> 0 0 0 0 1 1 0 0

The function should return 12 (1 1 0 0)

- **get_nbits_from_pos**
  ---
  a. Read number num from user.
  b. Read number n from user.
  c. Read number pos from user.
  d. Fetch n number of bits from given position 'pos' (starting from LSB) of num and return the decimal value of it.
  If num is 12, n is 3 and pos is 4
  7 6 5 4 3 2 1 0
  ---
  12 -> 0 0 0 0 1 1 0 0

  The function should return 3 (0 1 1).

- **replace_nbits_from_pos**
  ---
  a. Read number num from user.
  b. Read number n from user.
  c. Read number pos from user.
  d. Read number val from user.
  e. Fetch n number of bits from LSB of val.
  f. Place those fetched bits from pos positionth bit of num and return new value of num.
  If num is 12, n is 3, pos is 4 and val is 20
  7 6 5 4 3 2 1 0
  20 -> 0 0 0 1 0 1 0 0
  ---
  10 -> 0 0 0 0 1 1 0 0
  ---
  return value-> 0 0 0 1 0 0 0 0
  So function should return 16 (1 0 0 0 0).

- **toggle_bits_from_pos**
  ---
  a. Read number num from user
  b. Read number n from user
  c. Read number pos from user
  d. Invert the n number of bits from pos positionth bit of num.
  e. Return the new value of num
  If num is 10, n is 3, and pos is
  7 6 5 4 3 2 1 0
  ---
  10 -> 0 0 0 0 1 0 1 0
  ---

return value -> 0 0 1 1 0 0 1 0
So the function should return 50 (0 0 1 1 0 0 1 0)

- **print_bits**
  ---
  a. Read number num from user.
  b. Read number n from user.
  c. Do error checking
  -> If n is greater than integer size, assign n value as sizeof integer.
  d. Print n number of bits of num from LSB end.
  If num is 10 and n is 12, then print last 12 bits of binary representation of 10.
  The output should be -> 0 0 0 0 0 0 0 0 1 0 1 0

Pr-requisites:-
- Functions
- Bitwise Operators

Objective: -
- To understand the concept of
  - Bitwise Operators

Inputs: -
An integer N, No of bits B and Position P

Sample execution: -
Test Case 1:
user@emertxe] ./bit_ops
Select bit operation from below list
1. get_nbits
2. set_nbits
3. get_nbits_from_pos
4. set_nbits_from_pos
5. toggle_bits_from_pos
6. print_bits
Enter your choice: 3
Enter num : 12
Enter n : 3
Enter pos : 5
Value at 12[5:3] -> 1
Do you want to continue(y/Y) : n

Assignment 15

Read int i, Read 0 ≤ a ≤ b ≤ 31. Read an int n, put the (b-a+1) lsb's of n into i[b:a]

Description:

- Read number n from user.
- Read number i from user.
- Read number a from user(0 < a < 31)
  - Do error checking
  - Check a is within limit or not.
- Read number b from user(a < b < 31)
- Do error checking
  - Check b is within limit or not.
- Call set_nbits_from_pos function by passing i, b - a + 1, b and n as arguments.
  - set_nbits_from_pos(i, b - a + 1, b, n);
- Print the new value of i.
- Prompt for continue option.

Example:
If n = 11 = $(0\ 0\ 0\ 0\ 1\ 0\ 1\ 1)_2$
i = 174 = $(1\ 0\ 1\ 0\ 1\ 1\ 1\ 0)_2$
a = 3,
b = 5
then
New 'i' value = $(1\ 0\ 0\ 1\ 1\ 1\ 1\ 0)_2 = 158$

Pr-requisites:-
- Loops
- Bitwise Operators
- Type Modifiers
- Functions

Objective: -
- To understand the concept of
  - Functions
  - Bitwise Operators

Inputs: -
Integers N, I, A, B

Sample execution: -
Test Case 1:
user@emertxe] ./bit_ops
Enter the value of 'n' : 11
Enter the value of 'I' : 174
Enter the value of 'a' : 3
Enter the value of 'b' : 5
The binary form of 'n' : 00000000 00000000 00000000 00001011
The binary form of 'I' : 00000000 00000000 00000000 10101110
The new binary form of 'I' : 00000000 00000000 00000000 10011110
Want to continue [yY / nN] : n

Assignment 16

Read an int, & a no. n. Circular right & left shift the int by n.

Description:

- Read a number num from user.
- Read a number n from user.
- Select the option among
  - Circular right shift
  - Circular left shift
- Pass num and n to the corresponding functions.
- Shift num, n times (either left or right, depends on function).
- While shifting the shifted bits should get replaced at the alternate end.
  - For right shifting, the shifted bits should come at left most side.
  - For left shifting, the shifted bits should come at right most side.
- Return the new number from the function.
- Print the new number.
  Example:
  If num is 12, and n is 3, in circular_right_shift fuction
  12 -> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
  o/p -> 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

Pr-requisites:-

- Loops
- Bitwise Operators
- Type Modifiers
- Functions

Objective: -

- To understand the concept of
  - Functions
  - Bitwise Operators

Inputs: -

Integers N, S

Sample execution: -

Test Case 1:
user@emertxe] ./bit_ops
Enter num: 12
Enter n : 3
Possible operations
1. circular right shift

2. circular left shift
Enter your choice: 1
The binary form of number : 00000000 00000000 00000000 00001100
After circular right shifting by 3
The bianry form of number : 10000000 00000000 00000000 00000001
Want to continue [yY / nN] : y

Assignment 17

To write a program to swap two variables by using pass by reference method

Description:

- Read number a from user.
- Read number b from user.
- Call a swap function by passing address of both a & b.
- After swap function, a and b values should get swapped without using temprory variable and using bitwise operator.
- Prompt for continue option.

Pr-requisites:-
- Functions
- Pointers

Objective: -
- To understand the concept of
  - 2's Compliment and Bitwise Operators

Inputs: -
2 Integers N1 and N2

Sample execution: -
Test Case 1:
user@emertxe] ./swap
Enter N1: 10
Enter N2: 20
After swapping,
N1 = 20
N2 = 10

Assignment 18

To find the average of n numbers by taking input in 3 different ways.

Description:

- Ist method
  - Read number n from user.
  - Read n numbers from user.
  - Calculate average of entered numbers and print it on the screen.

- IInd method
  - Pass the numbers (for which average to be calculated) through command line.
  - Collect them in main
  - Calculate the average of them.

  Hint : Use atoi function to convert string to integers.

  IIIrd method
  - Pass the numbers (for which average to be calculated) through environment variables.
  - Collect them in main
  - Calculate the average of them.

  Hint :
  Use atoi function to convert string to integers.
  Use strtok to split the strings.

  Example : From shell export a variable which contain numbers.
  export arr="1 2 3 4"

  In main program receive this by third argument of main. Extract numbers from string and calculate their average.

Pr-requisites:-
- Main Prototypes
  Arrays and Pointers

Objective: -
- To understand the concept of
  - Arrays and pointers

Inputs: -
Array on N integers, N

Sample execution: -
Test Case 1:
user@emertxe] ./avg
Select the method to calculate average:
1. Scan from keyboard.
2. Proceed with environment variable arguments.
Enter you choice : 1
Enter the total number of integers : 4
Enter the elements : 3 5 10 7
The average of entered numbers : 6.25

Test Case 2:
user@emertxe] ./avg 3 5 10 7
Select the method to calculate average:
1. Scan from keyboard.
2. Proceed with provided command line arguments.
3. Proceed with environment variable arguments.

Enter you choice : 2
The average of entered numbers : 6.25

Test Case 3:
user@emertxe] export arr="1 2 3 4"
user@emertxe] ./avg
Select the method to calculate average:
1. Scan from keyboard.
2. Proceed with environment variable arguments.
Enter you choice : 2
The average of entered numbers : 6.25

Assignment 19

WAP to find factorial for given number using recursive method and without using any other function than main function

Description:

- Read number n from user.
- Validate the given number
- Call main function from main for calculating factorial.
- Prompt for continue option.

Pr-requisites:-
- Storage Classes
- Recursions

Objective: -
- To understand the concept of
  - Recursion and static keyword

Inputs: -
Integers N

Sample execution: -

Test Case 1:
user@emertxe] ./factorial
Enter the value of N: 7
Factorial of 7 is : 5040

Test Case 2:
user@emertxe] ./factorial
Enter the value of N: 5
Factorial of 5 is : 120

Test Case 3:
user@emertxe] ./factorial
Enter the value of N: -1
Invalid Input

Assignment 20

WAF for pre and post increment and passing int pointer as their parameter

Description:

- Pass a number 'N' from command line.
- If no numbers are entered through command line, read from user.
- Read the type of increment from user like menu drive.
  - post increment
  - pre increment
- Call the corresponding functions
- Return values of the functions should exactly behave as ++num and num++ operations
- We should use ++ operator in this program.
- With the help of bitwise operator we increment the given value.

  Example: -
  If num is 5,

  i = pre_increment(&num);
  printf("i = %d\n num = %d\n", i, num);

  should print,
  i = 6, num = 6

  If num is 5,

  i = post_increment(&num);
  printf("i = %d\n num = %d\n", i, num);

  should print,
  i = 5, num = 6

Pr-requisites:-

- Bitwise operators
- pointers

Objective: -

- To understand the concept of
  - To understand the concept of pre and post increment using bitwise

Inputs: -

Integer N

Sample execution: -

Test Case 1:
user@emertxe] ./inc
Enter the value of N : 5
Enter the operation :
1. Pre-Increment
2. Post-Increment
Choice : 2
After post increment i = 5, num = 6.

Assignment 21

WAP to generate fibbonacci numbers <= 'n' using recursions

Description:

- In mathematics, the Fibonacci numbers or Fibonacci sequence are the numbers in the following integer sequence
  1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 . . .
  OR
  0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 . . .
- By definition, the first two numbers in the Fibonacci sequence are either 1 and 1, or 0 and 1, depending on the chosen starting point of the sequence, and each subsequent number is the sum of the previous two.

Pr-requisites:-

- Loops
- Arithmetic Operators
- Recursions

Objective: -

- To understand the concept of
  - Recursive Functions

Inputs: -

Integers N

Sample execution: -

Test Case 1:
user@emertxe] ./fibbonacci
Enter the value of N: 8
Fibbonacci series using recursive functions is
0, 1, 1, 2, 3, 5, 8

Test Case 2: Positive Numbers
user@emertxe] ./fibbonacci_series
Enter the value of N: 10
Fibbonacci series using recursive functions is
0, 1, 1, 2, 3, 5, 8

Test Case 3: Negative Number
user@emertxe] ./fibbonacci_series
Enter the value of N: -21
Fibbonacci series using recursive functions is
0, 1, -1, 2, -3, 5, -8, 13, -21

Test Case 4: Negative Number
user@emertxe] ./fibbonacci_series
Enter the value of N: -13
Fibbonacci series using recursive functions is
0, 1, -1, 2, -3, 5, -8, 13

Assignment 22

To write a program to count number of characters, words and lines, entered through stdin buffer

Description:

- Read characters from user till EOF
- If EOF received, print the character count, word count, and line count.

Pr-requisites:-

- Non formated function like getchar()

Objective: -

- To understand the concept of
  - Standard output

Inputs: -

A string with Ctrl-D

Sample execution: -

Test Case 1:

```
user@emertxe] ./my_wc
Hello world
Dennis  Ritchie
Linux

Character count  : 33
Line count       : 3
Word count       : 5
```

Test Case 2:

```
user@emertxe] ./my_wc
Hello                world
Dennis      Ritchie
Linux

Character count  : 39
Line count       : 3
Word count       : 5
```

Assignment 23

Replace each string of one or more blanks by a single blank.

Description:

- Input string:
  Pointers are sharp knives.
- Output String:
  Pointers are sharp knives.

Pr-requisites:-
- Functions
- Pointers
- Dynamic Arrays

Objective: -
- To understand the concept of
  - Functions, Arrays, and Pointers

Inputs: -
String with multispaces between words

Sample execution: -
Test Case 1:
user@emertxe] ./remove_multi_space
Enter the string with more spaces in between two words
Pointers are sharp knives.
String with reduced space:
Pointers are sharp knives.

Assignment 24

Implement a my_cp() function

Description:

- Take two file names through command line
- Do error checking.
  - Check command line args
  - Check file open status
- Open first file in read mode
- Open second file in write mode
- Copy first file contents into second file.

Pr-requisites:-
- file operation functions like fopen, fgets etc

Objective: -
- To understand the concept of
  - File handling in C

Inputs: -

Source File and a Destination File

Sample execution: -

Test Case 1:

user@emertxe] ./my_cp file1.txt file2.txt
user@emertxe] ls
file1.txt **file2.txt**
user@emertxe]

Test Case 2:

user@emertxe] ./my_cp file1.txt
Destination file missing

Assignment 25

WAP to concatenate two files

Description:

- Receive file names through command line.
- Do error checking.
  - cmd line args
  - fopen status
- if cmd line args are ./a.out
  - then print stdin buffer contents into stdout buffer.
- If cmd line args are ./a.out file1 file2
  - then print file1 and file2 contents to stdout.
- If cmd line args are ./a.out file1 file2 file3
  - then print file1 and file2 contents into file3.
- Use a common function for doing file copy.

Pr-requisites:-

- file operation functions like fopen, fgets etc

Objective: -

- To understand the concept of
  - File handling in C

Inputs: -

Source Files, Destination File

Sample execution: -

Test Case 1:

user@emertxe] ./my_cat
Hello
Hello

Test Case 2:

user@emertxe] ./my_cat file1.txt
<contents of file1.txt>
user@emertxe]

Test Case 2:
user@emertxe] ./my_cat file1.txt file2.txt file3.txt
<copy the contents of file1.txt and file2.txt into file3.txt>
user@emertxe]

Assignment 26

To define a macro SIZEOF(x), where x is a variable, without using sizeof operator

Description:

- Treat address of x and address of x + 1as characters address.
- Both addresses difference will be sizeof x

Pr-requisites:-
- Macros
- Pointers

Objective: -
- To understand the concept of
  - Macros in preprocessing

Inputs: -
None

Sample execution: -
Sample execution: -
Test Case 1:
user@emertxe] ./sizeof
Size of int : 4 bytes
Size of char : 1 byte
Size of float : 4 bytes
Size of double : 8 bytes
Size of unsigned int : 4 bytes
Size of long int : 8 bytes

Assignment 27

To define a macro swap (t, x, y) that interchanges two arguments of type t

Description:

- Implement swap concept with the help of macro
- The type of arguments to swap will be passed as t
- swap (int, x, y) where x and y are of types int.

Pr-requisites:-

- Macros

Objective: -

- To understand the concept of
  - Macros in preprocessing

Inputs: -

Integers N1 and N2

Sample execution: -

Test Case 1:
user@emertxe] ./swap
Enter the num1 : 10
Enter the num2 : 20

After Swapping :

num1 : 20
num2 : 10

Assignment 28

WAP to Implement the student record using array of strutures.

Description:

- Implemet an student record which will contain roll no., Name of student, subject marks, average and grade.
- Read number of students and declare array of structure with no.of students, then read number of subjects and n subject names.
- Read the particulars for each student.
  - Roll.no
  - Name.
  - Score of each subject
- Calculate the average. Based on that decide the grade.

Pr-requisites:-

- Structures.

Objective:

- To understand the concept of
  - Structures

- Pointers

Sample execution:

Test Case 1:

user@emertxe] ./student_record
Enter no.of students : 2
Enter no.of subjects : 2
Enter the name of subject 1 : Maths
Enter the name of subject 2 : Science
----------Enter the student datails----------
Enter the student Roll no. : 1
Enter the student 1 name : Nandhu
Enter Maths mark : 99
Enter Science mark : 91
----------Enter the student datails----------
Enter the student Roll no. : 2
Enter the student 2 name : Bindhu
Enter Maths mark : 88
Enter Science mark : 78
----Display Menu----
1. All student details
2. Particular student details
Enter your choice : 2
----Menu for Particular student----
1. Name.
2. Roll no.
Enter you choice : 1
Enter the name of the student : Nandhu

Roll No.	Name	maths	science	Average	Grade
1	Nandhu	99	91	95	A

Do you want to continue to display(Y/y) : n
Do you want to continue(Y/y) : n

Assignment 29

WAP to demote the type of given float number to integer using bitwise operators and bitfields.

Description:

- Implement type demotion for float without using typecasting.
- Read a float number from user.
- Ex : num = 12.34
- IEEE format of 12.34

	S	Expo	Mantissa
12.34 -	0	10000010	10001010111000010100

- Exponent value is 130.
- Subtract Exp value with 127. No.of bits = Exp – 127.
- No.of bits = 130 – 127 , No.of bits = 3.
- Extarct only 3 bits from matissa and merge with 1.
- Mantissa – **100**01010111000010100

- 1 | 100 -> 1100 binary formet of 12. Interger part of 12.34 is 12.

More Examples:

4.56 - 0 | 10000001 | 00100011110101110000101
129 - 127 = **2** 1|00 = 4

6.2 - 0 | 10000001 | 10001100110011001100110010
129 - 127 = **2** 1|10 = 6

- **Note : Don’t use %d format specifier to print the integer part**

Pre-requisites:
- Bitwise operators.
- Bit-fields

Objective:
- To understand the concept of
  - Bit fields

Inputs:
- Read float value.

Sample execution: -

Test Case 1:
user@emertxe] ./demoted_int
Enter any float numer : 23.76
Demoted value is 23.000000
Test Case 2:
user@emertxe] ./demoted_int
Enter any float numer : -3.76
Demoted value is -3.000000
Test Case 3:
user@emertxe] ./demoted_int
Enter any float numer : 0.76
Demoted value is 0.000000
Test Case 4:
user@emertxe] ./demoted_int
Enter any float numer : 1.76
Demoted value is 1.000000

Assignment 30

WAF read_int to read an integer

Description:

- Read_int function should exactly behave like scanf("%d", &i)
- Implement read_int fuction without using scanf function.

Pr-requisites:-

- Functions
- Pointers

Objective: -

- To understand the concept of
  - Functions and Pointers

Inputs: -

An integer

Sample execution: -

Test Case 1:
user@emertxe] ./read_int
Enter a number: 212
The number is 212
Test Case 2:
user@emertxe] ./read_int
Enter a number: -212
The number is -212
Test Case 3:
user@emertxe] ./read_int
Enter a number: +212
The number is 212
Test Case 4:
user@emertxe] ./read_int
Enter a number: 212-
The number is 212
Test Case 5:
user@emertxe] ./read_int
Enter a number: as212
The number is 0
Test Case 6:
user@emertxe] ./read_int
Enter a number: 21267jk
The number is 21267
Test Case 7:
user@emertxe] ./read_int
Enter a number: */212
The number is 0

Assignment 31

Print bits of float & double. Check IEEE std.

Description:

- Read the choice from user.
  - float or double
- Read the fractional number from user.
- Print the bits of the entered number.

Pr-requisites:-

- Loops
- Bitwise Operators
- Type Casting
- Functions
- Pointers

Objective: -

- To understand the concept of
  - Type Casting
  - Functions
  - Type casting on Pointers
  - IEEE Standard representation

Inputs: -

A real number

Sample execution: -

Test Case 1:

user@emertxe] ./print_fd_bits
Enter the Choice:
1. float
2. double
Choice: 1
Enter the float value: 2.3
Sign Mantissa Exponent
---
0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1

Assignment 32

WAP to generate a n*n magic square (n is odd +ve no.)

Description:

- In recreational mathematics, a magic square is an arrangement of distinct numbers (i.e. each number is used once), usually integers, in a square grid, where the numbers in each row, and in each column, and the numbers in the main and secondary diagonals, all add up to the same number
- A magic square has the same number of rows as it has columns, and in conventional math notation, "n" stands for the number of rows (and columns) it has.
- Thus, a magic square always contains n2 numbers, and its size (the number of rows [and columns] it has) is described as being "of order n". Example: if n = 3, the magic square

8	1	6
3	5	7
4	9	2

- Read an odd number n from user.
  - Do error checking.
  - Check the number is odd or not.
- If not, continue step a.
- Create an n X n matrix.
- Insert 1 to (n * n) numbers into matrix.
- Arrange the numbers in such a way that, adding the numbers in any direction, either rowwise column wise or diagonal wise, should result in same answer.

Pr-requisites:-

- Loops
- Arrays
- Pointers

Objective: -

- To understand the concept of
  - 2D Arrays
  - Pointers on 2d arrays

Inputs: -

An integer N

Sample execution: -

Test Case 1:

user@emertxe] ./magic_square
Enter a number: 3
8 1 6
3 5 7
4 9 2
Do you want to continue (Y/y) : N

Assignment 33

To take 8 consecutive bytes in memory. Provide a menu to manipulate or display the value of variable of type t (input) & indexed at i (i/p)

Description:

- Allocate 8 consecutive bytes in memory
- Provide a display menu
  - 1. Add element
  - 2. Remove element
  - 3. Display element
  - 4. Exit from the program
- It should allow to add elements of different data types which data type size is less than or equal to 8

Pr-requisites:-
- Pointers
- Dynamic memory allocation

Objective: -
- To understand the concept of
  - Dynamic memory allocation

Inputs: -

Integers N1 and N2

Sample execution: -

Sample execution: -
Test Case 1:
user@emertxe] ./mem_manager
Menu :
1. Add element
2. Remove element
3. Display element
4. Exit from the program

Choice ---> 1

Enter the type you have to insert:
1. int
2. char
3. float
4. double

Choice ---> 2

Enter the char : k
1. Add element
2. Remove element
3. Display element
4. Exit from the program

```
Choice ---> 3
------------------------
0 -> k
------------------------

1. Add element
2. Remove element
3. Display element
4. Exit from the program

Choice ---> 1

Enter the type you have to insert:
1. int
2. char
3. float0 -> k
------------------------

1. Add element
2. Remove element
3. Display element
4. Exit from the program

Choice ---> 1

Enter the type you have to insert:
1. int
2. char
3. float
4. double

Choice ---> 1

Enter the int : 10

1. Add element
2. Remove element
3. Display element
4. Exit from the program

Choice ---> 3
------------------------
0 -> 1   (char)
1 -> 10  (int)
------------------------
1. Add element
2. Remove element
3. Display element
4. Exit from the program

Choice ---> 4
```

Assignment 34

Implement getword, atoi, itoa functions

Description:

- int getword(char *word)
  - Create a function named getword.
  - Function should read a string from user and store them in the char address.
  - Function should return the length of word.
- *int atoi(const char *s)*
  - The function will recieve a string and covert the number stored in the string into exact integer number.
  - Return the number.
- int itoa(int n, char *s)
  - Convert integer n into a string and store the string in s.
  - Return the length of string from the function.

Pr-requisites:-
- Functions
- Pointers

Objective: -
- To understand the concept of
  - Functions and Pointers

Inputs: -

String, String and Integer

Sample execution: -

Test Case 1:
user@emertxe] ./getword
Enter a word: Hello
You entered Hello
Test Case 2:
user@emertxe] ./getword
Enter a word: Hello World
You entered Hello

Test Case 3:
user@emertxe] ./atoi
Enter a numeric string: 12345
String to integer is 12345
Test Case 4:
user@emertxe] ./atoi
Enter a numeric string: -12345
String to integer is -12345
Test Case 5:
user@emertxe] ./atoi
Enter a numeric string: +12345
String to integer is 12345

Test Case 6:
user@emertxe] ./atoi
Enter a numeric string: +-12345
String to integer is 0
Test Case 7:
user@emertxe] ./atoi
Enter a numeric string: 12345-
String to integer is 12345
Test Case 8:
user@emertxe] ./atoi
Enter a numeric string: abcd12345
String to integer is 0
Test Case 9:
user@emertxe] ./atoi
Enter a numeric string: 12345abcd
String to integer is 12345

Test Case 10:
user@emertxe] ./itoa
Enter a numeric string: 1234
Integer to string is 1234

Assignment 35

Variance calculation with static arrays & with dynamic arrays

Description:

- In probability theory and statistics, variance measures how far a set of numbers is spread out. A variance of zero indicates that all the values are identical. Variance is always non-negative: a small variance indicates that the data points tend to be very close to the mean (expected value) and hence to each other, while a high variance indicates that the data points are very spread out around the mean and from each other.

  Example:

x(input)	D = X - Mean	D^2
9	-11	121
12	-8	64
15	-5	25
18	-2	4
20	0	0
22	2	4
23	3	9
24	4	16
26	6	36
31	11	121
Sum = 200		Sum = 400

  Mean = (sum of x) / size
  where : size = Number of items in the input
  Formula to calulate the variance:

sigma = (sum of D^2) / size

Pr-requisites:-

- Functions
- Pointers
- Static Arrays

Objective: -

- To understand the concept of
  - Functions and Pointers
  - Static Functions

Inputs: -

Array of N Integers, N

Sample execution: -

Test Case 1:

user@emertxe] ./variance
Enter the 10 elements:
[0] -> 9
[1] -> 12
[2] -> 15
[3] -> 18
[4] -> 20
[5] -> 22
[6] -> 23
[7] -> 24
[8] -> 26
[9] -> 31
The Variance of the entered numbers is 40

Assignment 36

Read n & n floats in a float array ’store’. Print the values in sorted order without modifying or copying ’store’

Description:

- Read the float elements from user and store them into an array.
- Run a loop for printing the float elements in sorted order.

Pr-requisites:-

- Functions
- Arrays
- Pointers

Objective: -

- To understand the concept of
  - Functions, Arrays, and Pointers

Inputs: -

Array on N floats, N

Sample execution: -

Test Case 1:

user@emertxe] ./sort

Enter the count of float elements need to store: 5

[0] = 10.0

[1] = 1.2

[2] = 3.2

[3] = 8.2

[4] = -1.4

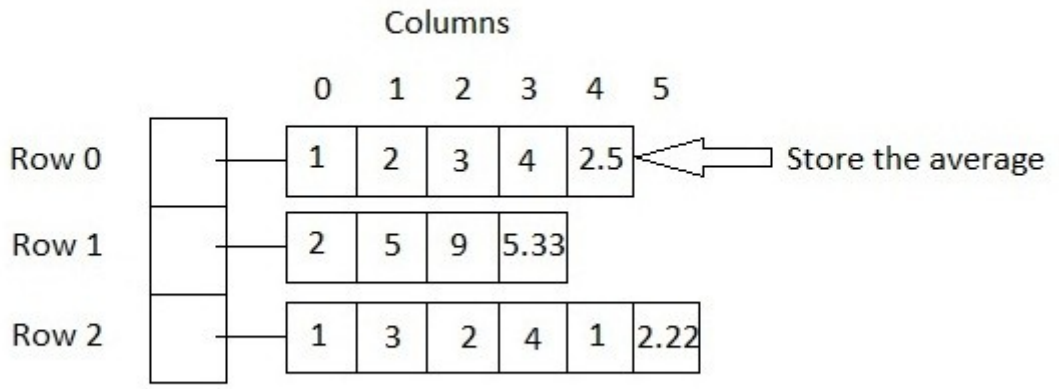
After sorting: -1.4 1.2 3.2 8.2 10.0

Assignment 37

WAP to implement fragments using Array of Pointers (First Static Second Dynamic).

Description:

- Implement fragments using array of pointers.
- Rows are static and columns are dynamic. Fixed no.of rows and columns will vary for each row.
- Example:
- Read no.of rows from user and allocate the memory statically for rows.
- Read no.of columns for each row and allocate the memory dynamically.
- Let us Assume, Row = 3.
- Row[0] = 4 columns, Row[1] = 3 columns and Row[2] = 5 columns.
- While allocating the memory for columns you have allocate for no.of columns + 1 dynamically.
- After that read the values from user and calculate the average for each row seperatly and store that average in that extra memory block which you added while allocating the memory.
- Example is given below.



- Then sort the array based on the average.
- Print the output on the screen.

Sample Execution:

Test case 1:

user@emertxe] ./fragments
Enter no.of rows : 3
Enter no of columns in row[0] : 4
Enter no of columns in row[1] : 3
Enter no of columns in row[2] : 5
Enter 4 values for row[0] : 1 2 3 4
Enter 3 values for row[1] : 2 5 9
Enter 5 values for row[2] : 1 3 2 4 1

Before Sorting output is:
1 2 3 4 2.5
2 5 9 5.33
1 3 2 4 1 2.22

After Sorting output is:
1 3 2 4 1 2.22
1 2 3 4 2.5
2 5 9 5.33
Do you want to continue(Y/y) : N

Assignment 38

Read n & n person names of maxlen 32. Sort and print the names

Description:

- Read the N name from the user
- Sort it in alphabetical order

Pr-requisites:-

- Functions
- Static / Dynamic Arrays
- Pointers

Objective: -

- To understand the concept of
  - Functions, Arrays, and Pointers

Inputs: -

Array on N names, N

Sample execution: -

Test Case 1:

user@emertxe] ./sort
Enter the 5 names of length max 32 characters in each
[0] -> Delhi
[1] -> Agra
[2] -> Kolkata
[3] -> Bengaluru
[4] -> Chennai

The sorted names are:
[0] -> Agra
[1] -> Bengaluru
[2] -> Chennai
[3] -> Delhi
[4] -> Kolkata

Assignment 39

To implement strtok function

Description:

- Read string1 and string2 from user.
- Call my_strtok (string1, string2);
- Should treat string2 as delimitter in string1 and should return 1st field.
- If you call again my_strtok (NULL, string2), it should return second field in string1 treating string2 as delimitter.

Pr-requisites:-
- Storage Classes
- Strings
- Pointers

Objective: -
- To understand the concept of
  - Strings functions

Inputs: -
2 Strings

Sample execution: -
Test Case 1:
user@emertxe] ./my_strtok
Enter string1 = ::Bangalore;;::---Chennai:;Kolkata:;Delhi:-:Mumbai
Enter string2 = ;./-:
Tokens :
Bangalore
Chennai
Kolkata
Delhi
Mumbai

Assignment 40

Read a string and print it in reverse without storing in an array using recursive method OR non-recursive method.

Description:

- Read a string from user.
- Implement in both recursive and non-recursive methods.

- Without using static keyword and global variable.

Pr-requisites:-
- Recursion

Objective: -
- To understand the concept of
  - Recursion

Inputs: -
String

Sample execution: -
Test Case 1:
user@emertxe] ./str_rev
Enter a string : Hello World
Reverse string is : dlroW olleH

Assignment 41

Write an alternative version of squeeze(s1, s2) that deletes each character in s1 that matches any character in the string s2

Description:

- Read two strings s1 and s2 from user.
- Remove the characters in s1 that matches with s2.
- Input string:
  - string 1: Dennis Ritchie
  - string 2: Linux
- Output String:
  - After squeeze s1: Des Rtche

Pr-requisites:-
- Functions
- Pointers
- Arrays

Objective: -
- To understand the concept of
  - Functions
  - Pointers & strings
  - Arrays

Inputs: -
String1 and String2

Sample execution: -
Test Case 1:
user@emertxe] ./squeeze
Enter s1 : Dennis Ritchie
Enter s2 : Linux
After squeeze s1 : Des Rtche

Assignment 42

WAP to check your processor endianess

Description:

- Check the memory assigning order of variables for checking whether little endian or big endian machine
  Little Endian:

  Say *int i = 10*; gets stored in memory at

  1000
  1001
  1002
  1003

  where 1000 contains LSB of i and 1003 contains MSB of i

  char *ptr = &i; the ptr will now contain 1000

  Big Endian:

  Say *int i = 10*; gets stored in memory at

  1000
  1001
  1002
  1003

  where 1000 contains MSB of i and 1003 contains LSB of i

  char *ptr = &i; the ptr will now contain 1000

  Fetch the values pointed by ptr for knowing whether they stored as little endian or big endian format.

Pr-requisites:-
- Pointers

Objective: -
- To understand the concept of
  - Strings functions

Inputs: -
2 Strings

Sample execution: -
Test Case 1:
user@emertxe] ./endianness
Little Endian System

Assignment 43

WAP to implement binary search for all basic datatypes.

Description:

- Enter the length of array from command line.
- Read the type of data to search
- Initialise the array of mentioned type with variables read from user.
- Read the key element to search
- Sort the element in ascending order.
- Return the position of element in the array.

Pr-requisites:-

- Void pointer concepts
- Function pointer concepts
- Functions and arrays

Objective: -

- To understand the concept of
  - Function and void pointers

Inputs: -

Array on N elements, N, Data type T, Element to be searched K

Sample execution: -

Sample execution: -
Test Case 1:
user@emertxe] ./binary_search 4
1. Int
2. Char
3. Float
4. Double
5. String
Choice: 1
Enter 4 elements of type int
1. 1
2. 2
3. 7
4. 4
After sorting: 1 2 4 7
Enter the key element to search: 1
Output :
Search Element 1 is at position 0 of array.

Assignment 44

Implement calc_mean for all data types

Description:

- Enter the number of elements to store in the array through the command line
- Read the type of data to store
- Enter the elements into the array.
- Pass them to calc mean function
- Should return mean value.

Pr-requisites:-

- Functions
- Void Pointers
- Arrays

Objective: -

- To understand the concept of
  - Functions
  - Void Pointers
  - Arrays

Inputs: -

Array of N elements, No of elements N, Size Option T,

Sample execution: -

Test Case 1:

user@emertxe] ./clac_mean
1. int
2. short
3. float
4. double
Enter the choice: 1
Enter 5 elements
a[0] -> 10
a[1] -> 20
a[2] -> 30
a[3] -> 40
a[4] -> 50
The mean value: 25.00

Assignment 45

WAP to implement my_printf() function.

Description:

- int my_printf (const char *format, ...);
  - The printf() function prints output on standard output stream.

Return value
  - Upon successful return, these function return the number of charaters printed.

Pr-requisites:-
- Variadic Functions
- Pointers

Objective: -
- To understand the concept of
  - Variadic Function

Inputs: -
As passed to standard printf() function

Implement for:
1. int
2. char
3. string

Sample execution: -
Test Case 1:
user@emertxe] ./my_printf
Enter a number, char, string: 20, c, Hello
Output: 20 c Hello

Assignment 46

WAP to implement my_scanf() function.

Description:

- int my_scanf (const char *format, ...);
  - The scanf() function reads input from stdin (the standard input stream).

Return value
  - Upon successful return, these function return the number of input item successfully matched and assigned

Pr-requisites:-
- Variadic Functions

- Pointers

Objective: -
- To understand the concept of
  - Variadic Function

Inputs: -
As passed to standard scanf() function

Implement for:
1. int
2. char
3. string
4. float
5. double

Sample execution: -
Test Case 1:
user@emertxe] ./my_scanf
Enter a number: 20
You entered 20
Enter a string: Hii
You entered Hii

Assignment 47

WAP implement the solution for tower of hanoi

Description:

- Tower of hanoi consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.
- The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules :
  - Only one disk can be moved at a time.
  - Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack (i.e) a disk can only be moved if it is the uppermost disk on a stack.
  - No disk may be placed on top of a smaller disk.

Pr-requisites:-
- Recursions

Objective: -
- To understand the concept of
  - Recursions

Inputs: -
Number of disks N

Sample execution: -
Test Case 1:

user@emertxe] ./toh
Enter the number of disks N: 3

make the legal move from rod A to C
make the legal move from rod A to B
make the legal move from rod C to B
make the legal move from rod A to C
make the legal move from rod B to A
make the legal move from rod B to C
make the legal move from rod A to C

Assignment 48

WAP to take n and k (1 <= k <= 10) as inputs. Generate consecutive NRPS of length n using k distinct character (0 <= k <= 9)

Description:

- Suppose k = 3 (say taking 3 distinct characters).
  - Let 3 distinct characters be a , b, c.
- Suppose ‘n’ is the string length to be formed using ‘k’ distinct words.
  - Let n be 6
- The string should be formed in such a way that there should not be any consecutive repetitions of the strings.

Pr-requisites:-
- Strings
- Pointers
- Arrays

Objective: -
- To understand the concept of
  - String manupulations

Inputs: -
No of character C, Length of the string N and C disctint characters

Sample execution: -
Test Case 1:
user@emertxe] ./nrps
Enter the number characters C : 3
Enter the Lenght of the string N : 3
Enter 3 distinct characters : a b c
Possible NRPS is abcacb

Assignment 49

WAP to print all possible combinations of given string.

Description:

- Find and print all the possible combinations of given string.

Pr-requisites:-

- Strings
- Pointers
- Arrays

Objective: -

- To understand the concept of
  - String manupulations

Inputs:

- Read a string.

Sample execution: -

Test Case 1:
user@emertxe] ./combinations
Enter a string: abc
abc
acb
bca
bac
cab
cba
Do you want to continue (Y/y) : y
Sample execution: -
Test Case 1:
user@emertxe] ./combinations
Enter a string: abb
Error please enter distinct characters.
Do you want to continue(Y/y) : n

Assignment 50

WAP to find the product of given matrix.

Description:

- Read no.of rows and columns from user and allocate the memory dynamically using malloc or calloc (Assume Matrix A).
- Read the Matrix A from user.
- Then find the transpose of the given matrix and strore it in another dynamic array (Assume Matrix AT) .
- Find the product for matrix A and matrix AT.
- Let say Name of the matrix is A and no. Of rows = columns = 3.
  - Matrix A

1	2	3
1	2	3
1	2	3

- Matrix AT (Transepose of A)

1	1	1
2	2	2
3	3	3

- Final Result :
  - Result = A * AT

R = 1 2 3 / 1 2 3 / 1 2 3 x 1 1 1 / 2 2 2 / 3 3 3

R1 = [(1 * 1) + (2 * 2) + (3 * 3) (1 * 1) + (2 * 2) + (3 * 3) (1 * 1) + (2 * 2) + (3 * 3)]

Final Result matrix

14	14	14
14	14	14
14	14	14

Pre-requisites:
- 2D – Arrays / Pointers.
- DMA.

Objective:
- To understand the concept of
  - DMA
  - Double Pointers / 2D array.

Inputs:
- No.of rows, Columns and row * column values.

Sample output:

Test case1:
user@emertxe] ./transpose_product
Enter number of rows : 3
Enter number of columns : 3
Enter values for 3 x 3 matrix :
1 2 3
1 2 3
1 2 3

Transpose of given Matrix :
1 1 1
2 2 2
3 3 3

Product of two matrix :
14 14 14

```
14      14      14
14      14      14
Do you want to continue(Y/y)? :  N
```