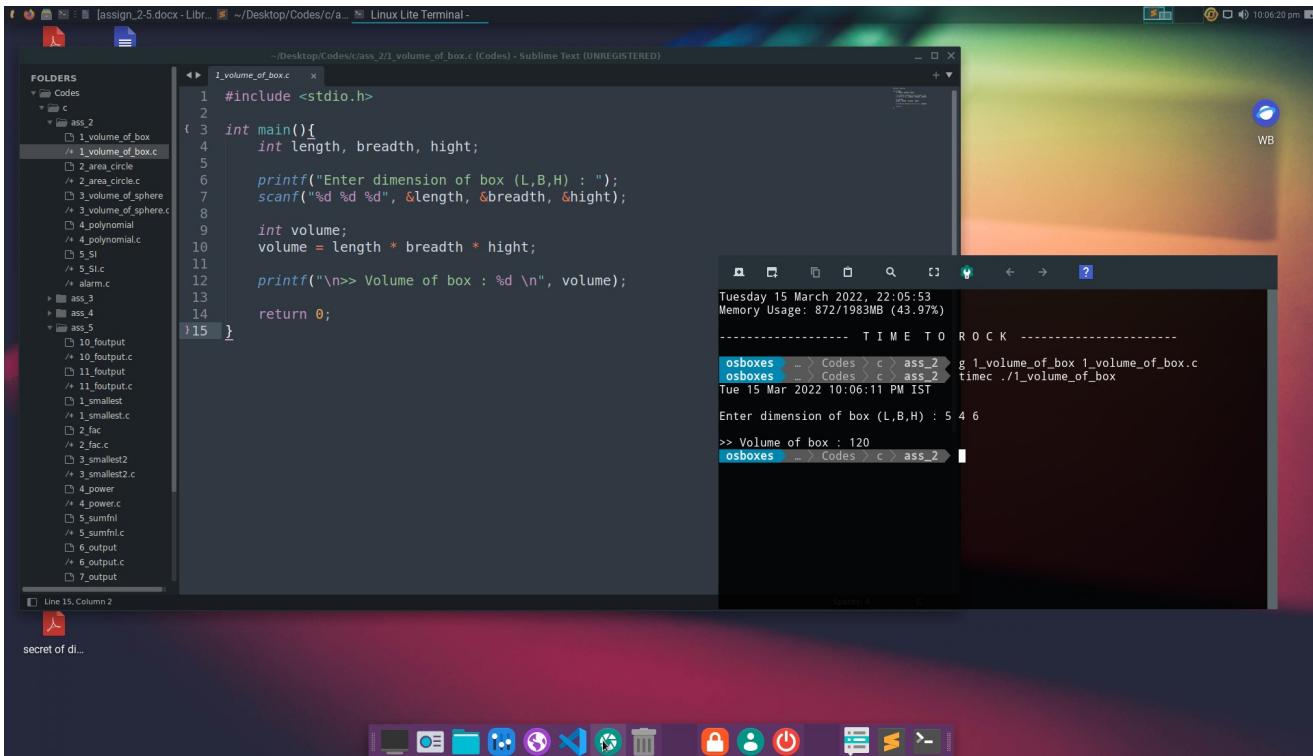


Roll no: 220350320070
Name: MD SAIF
PG-DAC March 2022
Module 01: Basic Programming Concepts

DAY 2

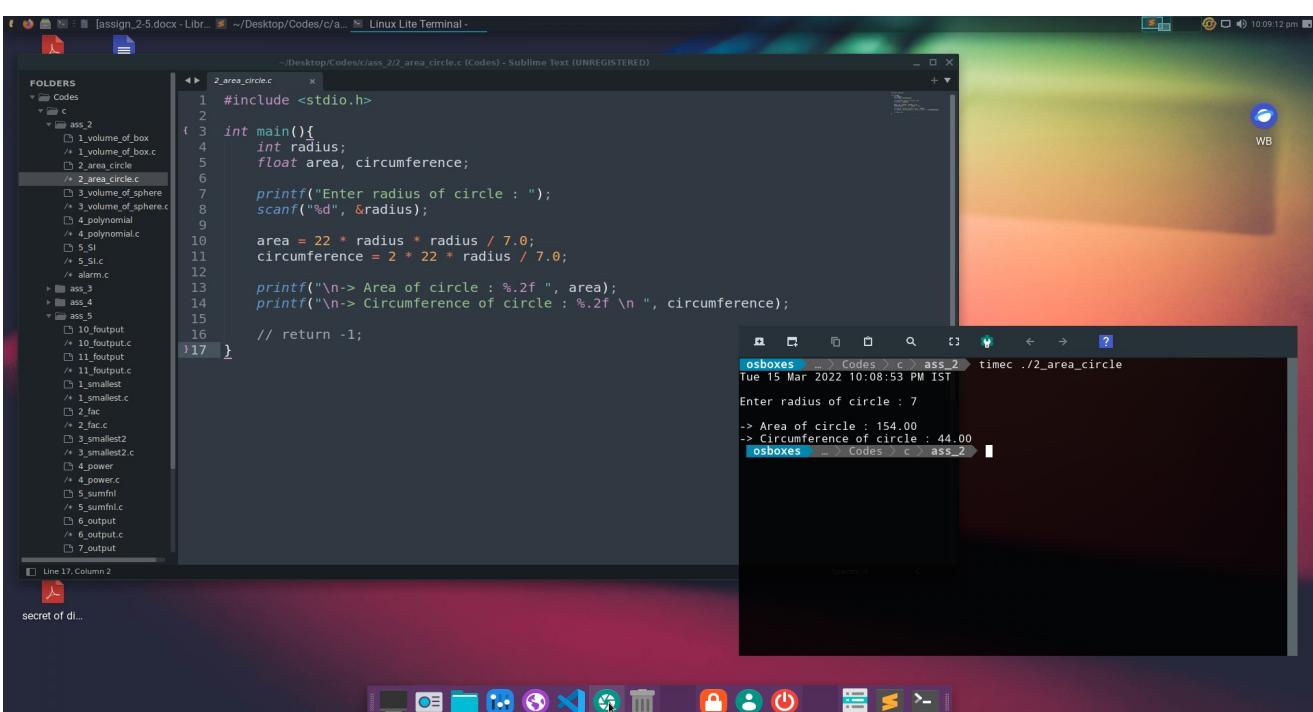
Solution 1:



The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window is titled 'Linux Lite Terminal' and shows the command 'timec ./1_volume_of_box' being run. The output shows the program asking for dimensions and calculating the volume of a box. The code editor window is titled '1_volume_of_box.c' and contains C code for calculating the volume of a box.

```
#include <stdio.h>
int main(){
    int length, breadth, height;
    printf("Enter dimension of box (L,B,H) : ");
    scanf("%d %d %d", &length, &breadth, &height);
    int volume;
    volume = length * breadth * height;
    printf("\n>> Volume of box : %d \n", volume);
    return 0;
}
```

Solution 2



The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window is titled 'Linux Lite Terminal' and shows the command 'timec ./2_area_circle' being run. The output shows the program asking for the radius of a circle and calculating its area and circumference. The code editor window is titled '2_area_circle.c' and contains C code for calculating the area and circumference of a circle.

```
#include <stdio.h>
int main(){
    int radius;
    float area, circumference;
    printf("Enter radius of circle : ");
    scanf("%d", &radius);
    area = 22 * radius * radius / 7.0;
    circumference = 2 * 22 * radius / 7.0;
    printf("\n-> Area of circle : %.2f ", area);
    printf("\n-> Circumference of circle : %.2f \n ", circumference);
    // return -1;
}
```

Solution 3

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for 'WB'. On the desktop, there is a Sublime Text window with two tabs open: '2_area_circle.c' and '3_volume_of_sphere.c'. The file '3_volume_of_sphere.c' contains the following C code:

```
#include <stdio.h>
#define PI (float)22/7

int main(){
    int radius=10;

    printf("Enter radius of sphere : ");
    scanf("%d", &radius);

    float volume;
    volume = (float)4/3 * PI * radius * radius * radius;

    printf("\n>> Volume of box : %.2f \n", volume);

    return 0;
}
```

Below the Sublime Text window is a terminal window titled 'osboxes' showing the output of the program:

```
osboxes ~ > Codes > c > ass_2 timecc ./3_volume_of_sphere
Tue 15 Mar 2022 10:12:33 PM IST
Enter radius of sphere : 7
>> Volume of box : 1437.33
osboxes ~ ... > Codes > c > ass_2
```

Solution 4

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for 'WB'. On the desktop, there is a Sublime Text window with three tabs open: '2_area_circle.c', '3_volume_of_sphere.c', and '4_polynomial.c'. The file '4_polynomial.c' contains the following C code:

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

int main(){
    int x, y;

    printf("Enter value of x : ");
    scanf("%d", &x);

    y = 6*pow(x, 5) + 5*pow(x, 4) - 4*pow(x, 3) + 3*pow(x, 2) + 2*x + 1;

    printf("\n-> y = 6x^5 + 5x^4 - 4x^3 + 3x^2 + 2x + 1 \n-> y = %d\n", y);

    return 0;
}
```

Below the Sublime Text window is a terminal window titled 'osboxes' showing the output of the program and some compilation errors:

```
osboxes ~ > Codes > c > ass_2 g 4_polynomial 4_polynomial.c
/usr/bin/ld: /tmp/cc7Q1SLM.o: in function `main':
4_polynomial.c:(.text+0x54): undefined reference to `pow'
/usr/bin/ld: 4_polynomial.c:(.text+0x79): undefined reference to `pow'
/usr/bin/ld: 4_polynomial.c:(.text+0xa3): undefined reference to `pow'
/usr/bin/ld: 4_polynomial.c:(.text+0xd1): undefined reference to `pow'
collect2: error: ld returned 1 exit status
osboxes ~ > Codes > c > ass_2 g 4_polynomial 4_polynomial.c -lm
osboxes ~ > Codes > c > ass_2 timecc ./4_polynomial
Tue 15 Mar 2022 10:16:44 PM IST

Enter value of x : 1
-> y = 6x^5 + 5x^4 - 4x^3 + 3x^2 + 2x + 1
-> y = 13
osboxes ~ > Codes > c > ass_2 timecc ./4_polynomial
Tue 15 Mar 2022 10:16:58 PM IST

Enter value of x : 3
-> y = 6x^5 + 5x^4 - 4x^3 + 3x^2 + 2x + 1
-> y = 1789
osboxes ~ > Codes > c > ass_2
```

Solution 5

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon labeled "WB". On the desktop, there is a Sublime Text window titled "2_area_circle.c" which contains C code for calculating the area of a circle. Below it is a terminal window titled "osboxes" showing the execution of a program to calculate simple interest.

Sublime Text Content:

```
1 #include <stdio.h>
2
3 int main(){
4     float SI, principle, rate, time;
5
6     printf("Enter principle : ");
7     scanf("%f", &principle);
8     printf("Enter rate : ");
9     scanf("%f", &rate);
10    printf("Enter time : ");
11    scanf("%f", &time);
12
13    SI = principle * rate * time /100;
14
15    printf("\n-> Simple Interest = %.2f\n", SI);
16
17    return 0;
18 }
```

Terminal Output:

```
osboxes ... > Codes > c > ass_2 > timec ./4_polynomial
Tue 15 Mar 2022 10:16:44 PM IST
Enter value of x : 1
-> y = 6x^5 + 5x^4 - 4x^3 + 3x^2 + 2x + 1
-> y = 13
osboxes ... > Codes > c > ass_2 > timec ./4_polynomial
Tue 15 Mar 2022 10:16:58 PM IST
Enter value of x : 3
-> y = 6x^5 + 5x^4 - 4x^3 + 3x^2 + 2x + 1
-> y = 1789
osboxes ... > Codes > c > ass_2 > g 5_SI 5_SI.c
osboxes ... > Codes > c > ass_2 > timec ./5_SI
Tue 15 Mar 2022 10:18:08 PM IST
Enter principle : 230
Enter rate : 12
Enter time : 32
-> Simple Interest = 883.20
osboxes ... > Codes > c > ass_2 >
```

DAY 3

solution 1

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window is titled 'Linux Lite Terminal' and shows the output of a program named '1_operators'. The code in the editor is as follows:

```
#include <stdio.h>
int main()
{
    int a, b;
    printf("Enter two numbers : ");
    scanf("%d%d", &a, &b);
    printf("\n%d + %d = %.2f\n", a, b, (float)a+b);
    printf("%d - %d = %.2f\n", a, b, (float)a-b);
    printf("%d * %d = %.2f\n", a, b, (float)a*b);
    printf("%d / %d = %.2f\n", a, b, (float)a/b);
    printf("%d %% %d = %d\n", a, b, a%b);
    return 0;
}
```

The terminal output shows the results for the input values 4 and 2.

solution 2

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window is titled 'Linux Lite Terminal' and shows the output of a program named '2_bitwise_operators'. The code in the editor is as follows:

```
#include <stdio.h>
int main()
{
    int a, b;
    printf("Enter two numbers : ");
    scanf("%d%d", &a, &b);
    // Bitwise Operators
    printf("\n~d = %d\n", a, ~a);
    printf("~d = %d\n", b, ~b);
    printf("%d ^ %d = %d\n", a, b, a^b);
    printf("%d & %d = %d\n", a, b, a&b);
    printf("%d | %d = %d\n", a, b, a|b);
    printf("%d >> %d = %d\n", a, b, a>>b);
    printf("%d << %d = %d\n", a, b, a<<b);
    return 0;
}
```

The terminal output shows the results for the input values 6 and 2.

Solution 3

The screenshot shows a Linux Lite desktop environment. In the foreground, a Sublime Text window displays a C program named `3_add_ass.c`. The code reads two integers from the user, adds them, and prints the result. Below the Sublime Text window is a terminal window titled "Linux Lite Terminal". The terminal shows the execution of the program and its output.

```
#include <stdio.h>
int main()
{
    int a, b;
    printf("Enter two numbers : ");
    scanf("%d%d", &a, &b);
    a += b;
    printf("\nAddition: a + b = %d\n", a);
    return 0;
}
```

```
osboxes:~/Desktop/Codes/c/ass_3$ g 2_bitwise_operators 2_bitwise_operators.c
osboxes:~/Desktop/Codes/c/ass_3$ timec ./2_bitwise_operators
Tue 15 Mar 2022 11:44:46 PM IST
Enter two numbers : 6 2
~6 = -7
~2 = -3
6 ^ 2 = 4
6 & 2 = 2
6 | 2 = 6
6 >> 2 = 1
6 << 2 = 24
osboxes:~/Desktop/Codes/c/ass_3$ g 3_add_ass 3_add_ass.c
osboxes:~/Desktop/Codes/c/ass_3$ timec ./3_add_ass
Wed 16 Mar 2022 12:27:31 AM IST
Enter two numbers : 5 3
Addition: a + b = 8
osboxes:~/Desktop/Codes/c/ass_3$
```

Solution 4

The screenshot shows a Linux Lite desktop environment. In the foreground, a Sublime Text window displays a C program named `4_maxOf2.c`. The code reads two integers from the user and prints the maximum of the two. Below the Sublime Text window is a terminal window titled "Linux Lite Terminal". The terminal shows the execution of the program and its output.

```
#include <stdio.h>
int main()
{
    int a, b;
    printf("Enter two numbers : ");
    scanf("%d%d", &a, &b);
    a>b ? printf("\nmaximum is %d\n", a) : printf("\nmaximum is %d\n", b);
    return 0;
}
```

```
osboxes:~/Desktop/Codes/c/ass_3$ g 2_bitwise_operators 2_bitwise_operators.c
osboxes:~/Desktop/Codes/c/ass_3$ timec ./2_bitwise_operators
Tue 15 Mar 2022 11:44:46 PM IST
Enter two numbers : 6 2
~6 = -7
~2 = -3
6 ^ 2 = 4
6 & 2 = 2
6 | 2 = 6
6 >> 2 = 1
6 << 2 = 24
osboxes:~/Desktop/Codes/c/ass_3$ g 3_add_ass 3_add_ass.c
osboxes:~/Desktop/Codes/c/ass_3$ timec ./3.add_ass
Wed 16 Mar 2022 12:27:31 AM IST
Enter two numbers : 5 3
Addition: a + b = 8
osboxes:~/Desktop/Codes/c/ass_3$ g 4_maxOf2 4_maxOf2.c
osboxes:~/Desktop/Codes/c/ass_3$ timec ./4_maxOf2
Wed 16 Mar 2022 12:28:21 AM IST
Enter two numbers : 5 3
maximum is 5
osboxes:~/Desktop/Codes/c/ass_3$
```

Day 4

Solution 1:

The screenshot shows a Linux desktop environment with a dark theme. On the left is a Sublime Text window displaying a C program named `1_positice_negetive.c`. The code checks if a float value is positive, negative, or zero. On the right is a terminal window titled "Linux Lite Terminal" showing the execution of the program. It prompts for a value, and the user enters 8, 0, and -4 respectively, with the output indicating each case.

```
#include <stdio.h>
int main()
{
    float val;
    printf("Enter value : ");
    scanf("%f", &val);

    if(val>0)
        printf("%.2f is positive\n", val);
    else if(val<0)
        printf("%.2f is negetive\n", val);
    else
        printf("Number is zero\n");

    return 0;
}
```

```
osboxes ... > Codes > c > ass_4 cd ./ass_4
osboxes ... > Codes > c > ass_4 g 1_positice_negetive 1_positice_negetive
Wed 16 Mar 2022 12:30:00 AM IST
Enter value : 8
8.00 is positive
osboxes ... > Codes > c > ass_4 g timec ./1_positice_negetive
Wed 16 Mar 2022 12:30:18 AM IST
Enter value : 0
Number is zero
osboxes ... > Codes > c > ass_4 g timec ./1_positice_negetive
Wed 16 Mar 2022 12:30:25 AM IST
Enter value : -4
-4.00 is negetive
osboxes ... > Codes > c > ass_4 g timec ./1_positice_negetive
```

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. On the left is a Sublime Text window displaying a C program named `2_greater_num.c`. The code reads three float values and prints the greatest one. On the right is a terminal window titled "Linux Lite Terminal" showing the execution of the program. It prompts for three values, and the user enters 8, 0, and -4 respectively, with the output indicating the greatest value in each case.

```
#include <stdio.h>
int main()
{
    float val1, val2, val3;
    printf("Enter 3 numbers : ");
    scanf("%f %f %f", &val1, &val2, &val3);

    float max = val1>val2 && val1>val3 ? val1 : val2>val1 && val2>val3 ? val2 : val3;
    printf("\n%.0f is greatest in all\n", max);

    return 0;
}
```

```
osboxes ... > Codes > c > ass_4 cd ./ass_4
osboxes ... > Codes > c > ass_4 g 1_positice_negetive 1_positice_negetive
Wed 16 Mar 2022 12:30:00 AM IST
Enter value : 8
8.00 is positive
osboxes ... > Codes > c > ass_4 g timec ./1_positice_negetive
Wed 16 Mar 2022 12:30:18 AM IST
Enter value : 0
Number is zero
osboxes ... > Codes > c > ass_4 g timec ./1_positice_negetive
Wed 16 Mar 2022 12:30:25 AM IST
Enter value : -4
-4.00 is negetive
osboxes ... > Codes > c > ass_4 g 2_greater_num 2_greater_num.c
osboxes ... > Codes > c > ass_4 g timec ./2_greater_num
Wed 16 Mar 2022 12:31:45 AM IST
Enter 3 numbers : 6 3 8
8 is greatest in all
osboxes ... > Codes > c > ass_4 g timec ./2_greater_num
Wed 16 Mar 2022 12:31:54 AM IST
Enter 3 numbers : 23 65 12
65 is greatest in all
osboxes ... > Codes > c > ass_4 g timec ./2_greater_num
```

solution 3

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for 'WIR'. Below it, a terminal window titled 'Linux Lite Terminal' is open, showing command-line output. The terminal window has a dark background with light-colored text. The output shows the execution of a C program named '3_smallest_num.c' to find the smallest number among three inputs (6, 3, 9), with the result '3 is smallest number'. The terminal window also shows other commands run previously, such as 'timec ./3_smallest_num' and 'Enter 3 numbers : 4 8 0'. In the bottom left, there is a file browser window titled 'assign_2-5.docx - LibreOffice - ~/Desktop/Codes/c/a...' showing a directory structure under 'Codes'. A Sublime Text editor window is also visible, displaying the source code for '3_smallest_num.c'. The code uses standard C syntax to read three float values from the user, compare them, and print the smallest one.

```
#include <stdio.h>
int main()
{
    float val1, val2, val3;
    printf("Enter 3 numbers : ");
    scanf("%f %f %f", &val1, &val2, &val3);
    float min;
    if(val1<val2 && val1<val3)
        min = val1;
    else if(val2<val1 && val2<val3)
        min = val2;
    else
        min = val3;
    printf("\n%.0f is smallest number\n", min);
    return 0;
}
```

solution 4

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for 'WIR'. Below it, a terminal window titled 'Linux Lite Terminal' is open, showing command-line output. The terminal window has a dark background with light-colored text. The output shows the execution of a C program named '4_find_output.c' to determine the output based on the value of variable 'i' (5 or 0). The result 'CDAC' is printed. The terminal window also shows other commands run previously, such as 'timec ./3_smallest_num' and 'Enter 3 numbers : 4 8 0'. In the bottom left, there is a file browser window titled 'assign_2-5.docx - LibreOffice - ~/Desktop/Codes/c/a...' showing a directory structure under 'Codes'. A Sublime Text editor window is also visible, displaying the source code for '4_find_output.c'. The code includes a conditional statement where 'i' is assigned a value of 5, and then prints either 'CDAC' or 'C Programming' based on the value of 'i'.

```
#include <stdio.h>
int main()
{
    int i;
    /* check condition is true or false check with
       (4,5,0)-> false output = CDAC
       (9,23,1)-> true output = C programming
       also (2,3,6)-> true output = C programming */
    if( a, b, c ) // in this case condition value taken from c i.e value of c take f
    {
        if(i==5,0,1)      // 5 willassing to var i and cond is true bcoz of 1
            printf("CDAC ");
        else
            printf("C Programming ");
        printf("%d\n", i);      // output 5
    }
    return 0;
}
```

solution 5

A screenshot of a Linux desktop environment. In the foreground, there is a terminal window titled "Linux Lite Terminal" with the command "timed ./5_find_output" and its output showing the result 216. In the background, there is a Sublime Text window displaying a C program named "5_find_output.c". The code calculates the value of r based on the formula $r = 12 + 12 * -18 / 7 \% 4 + 7 * 4$, which results in 216. The terminal window also shows the command "g 5_find_output 5_find_output" and the date "Wed 16 Mar 2022 12:41:32 AM IST".

```
#include <stdio.h>
int main()
{
    int a=12, b=18, c=7, d=4, result;
    result = a + a * -b/c % d + c * d;
    printf("%d\n", result);
    return 0;
}
```

solution 6

A screenshot of a Linux desktop environment. In the foreground, there is a terminal window titled "Linux Lite Terminal" with the command "timed ./6_find_output" and its output showing the result 216. In the background, there is a Sublime Text window displaying a C program named "6_find_output.c". The code contains a conditional statement where z is assigned 10 if y is equal to 10, and true1 is set to 1. The terminal window also shows the command "g 6_find_output 6_find_output" and the date "Wed 16 Mar 2022 12:43:38 AM IST".

```
#include <stdio.h>
int main()
{
    int y=10;
    int z = y+(y==10);
    // z = 10 + ( 10 == 10)      true1
    // z = 10 + (true)          true1
    // z = 10 + 1
    // z = 11
    printf("%d\n", z);
    return 0;
}
```

solution 7

A screenshot of a Linux desktop environment. At the top, there's a menu bar with icons for system settings and user accounts. Below the menu is a dock with various application icons. In the center, there's a terminal window titled "Linux Lite Terminal" showing command-line output. To its left is a code editor window titled "assign_2-5.docx - Libr..." showing C code for a program named "7_find_output.c". The code calculates a value based on user input and prints it. Below the code editor is a file browser showing a directory structure under "Codes". On the right side of the screen is a calculator application.

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
    int h = 9;
    int b = 5 * 2 + 2 * 3 < h*2 ? 5:3
    // b = 16<19 ? 5:3
    // b = 5
    printf("%d\n", b);
    return 0;
}
```

```
osboxes: ~ > Codes > ass_4 > ass_4 g 7_find_output 7_find_output.c
osboxes: ~ > Codes > ass_4 > ass_4 timec ./7_find_output
Wed 16 Mar 2022 12:45:28 AM IST
osboxes: ~ > Codes > ass_4
```

Calculator output: 216

solution 8

A screenshot of a Linux desktop environment, similar to the one above. It shows a terminal window, a code editor, and a calculator application. The terminal window displays the command-line output for a program named "8_find_output.c". The code in the editor checks if a variable x is 0 or 1 and prints "true" or "false" accordingly. The file browser on the left shows the project structure under "Codes".

```
#include <stdio.h>
int main()
{
    int x = 0;
    if(x++)
        printf("true\n");
    else if(x==1)
        printf("false\n");
    return 0;
}
```

```
osboxes: ~ > Codes > ass_4 > ass_4 g 8_find_output 8_find_output.c
osboxes: ~ > Codes > ass_4 > ass_4 timec ./8_find_output
false
osboxes: ~ > Codes > ass_4
```

Calculator output: 216

solution 9

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for a calendar. The desktop has a green and yellow gradient background. A Sublime Text window is open, showing a C program named `9_find_output.c`. The code contains a simple if-else statement to print "true" or "false" based on the condition `c>b`. To the right of the Sublime Text window is a terminal window titled "Linux Lite Terminal". The terminal shows the command `g 9_find_output 9_find_output.c` and its output, which is "false". Below the terminal is a calculator application.

```
#include <stdio.h>
int main()
{
    int a = 1, b = 2, c = 3;
    if (c>b)
        printf("true\n");
    else
        printf("false\n");
    return 0;
}
```

Solution 10

The screenshot shows a Linux desktop environment with a dark theme, similar to the previous one. In the top right corner, there is a system tray icon for a calendar. The desktop has a green and yellow gradient background. A Sublime Text window is open, showing a C program named `10_find_output.c`. The code performs various operations on variables `a`, `b`, `c`, and `d`, including incrementing and decrementing them and printing their values. To the right of the Sublime Text window is a terminal window titled "Linux Lite Terminal". The terminal shows the command `g 10_find_output 10_find_output.c` and its output, which includes "Kolkata", "Delhi", and "Bangalore". Below the terminal is a calculator application.

```
#include <stdio.h>
int main()
{
    int a = 1, b = -1, c = 0, d;
    d = ++a && ++b || c--; //after pass a = 2, b = 0, c = -1, d = 0
    // d = 2 && 0 || 0
    // d = 0 || 0
    // d = 0
    c=-1
    if(d)
        printf("Kolkata\n");
    else if(c) //c=-1 nonzero true
        printf("Delhi\n");
    else
        printf("Bangalore\n");
    printf("a=%d b=%d c=%d d=%d\n", a, b, c, d);
    return 0;
}
```

Day5

Solution 1:

The screenshot shows a Sublime Text window displaying a C program named `1_smallest.c`. The code uses nested if-else statements to find the minimum of three integers `a`, `b`, and `c`. It includes `#include <stdio.h>` and `int main()` blocks. Below the code editor is a terminal window titled "Linux Lite Terminal". The terminal shows the command `timec ./1_smallest` being run, followed by three test cases where the user enters values for `a`, `b`, and `c`, and the program outputs the smallest value. The terminal also shows the time taken for each run.

```
#include <stdio.h>
int main()
{
    int a, b, c, min;
    printf("Enter the three values : ");
    scanf("%d %d %d", &a, &b, &c);
    // with if-else ladder
    if(c>a && c>b){
        if(a>b)
            min = b;
        else
            min = a;
    }
    else if(a>b && a>c){
        if(b>c)
            min = c;
        else
            min = b;
    }
    else if(b>c && b>a){
        if(c>a)
            min = a;
        else
            min = c;
    }
    printf("%d is smallest\n", min);
    return 0;
}
```

```
osboxes ... > Codes > c > ass_5 > g 1_smallest 1_smallest.c
Wed 16 Mar 2022 01:07:23 AM IST
Enter the three values : 8 4 7
4 is smallest
osboxes ... > Codes > c > ass_5 > timec ./1_smallest
Wed 16 Mar 2022 01:07:30 AM IST
Enter the three values : 9 0 2
0 is smallest
osboxes ... > Codes > c > ass_5 > timec ./1_smallest
Wed 16 Mar 2022 01:07:37 AM IST
Enter the three values : 5 9 3
3 is smallest
osboxes ... > Codes > c > ass_5 > [REDACTED]
```

Solution 2

The screenshot shows a Sublime Text window displaying a C program named `2_fac.c`. The code calculates the factorial of a given number `num`. It includes `#include <stdio.h>` and `int main()` blocks. The code uses a while loop to multiply `fact` by `num` until `num` reaches 0. It handles negative numbers by breaking the loop. Below the code editor is a terminal window titled "Linux Lite Terminal". The terminal shows the command `timec ./2_fac` being run, followed by a test case where the user enters a number, and the program outputs the factorial. The terminal also shows the time taken for the execution.

```
#include <stdio.h>
int main()
{
    double num, fact=1, temp;
    printf("Enter a number : ");
    scanf("%lf", &num);
    temp=num;
    while(num){
        if(num<0)           //for negative numbers fact = 1
            break;
        fact = fact * num;
        num--;
    }
    printf("\nFactorial of %.0lf : %.0lf\n", temp, fact);
    return 0;
}
```

```
osboxes ... > Codes > c > ass_5 > g 2_fac 2_fac.c
Wed 16 Mar 2022 01:15:12 AM IST
Enter a number : 7
Factorial of 7 : 5040
osboxes ... > Codes > c > ass_5 > [REDACTED]
```

Solution 3

A screenshot of a Linux desktop environment. On the left, there is a file browser window showing a directory structure under 'Codes'. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'timec ./3_smallest2'. The terminal output shows the program being run and its results. On the right, there is a calculator application.

```
osboxes ... > Codes > c > ass_5 > g_2_fac_2_fac.c
osboxes ... > Codes > c > ass_5 > timec ./2_fac
Wed 16 Mar 2022 01:15:12 AM IST

Enter a number : 7
Factorial of 7 : 5040
osboxes ... > Codes > c > ass_5 > g_3_smallest2_3_smallest2.c
osboxes ... > Codes > c > ass_5 > timec ./3_smallest2
Wed 16 Mar 2022 01:16:40 AM IST

Enter the three numbers : 5 3 8
3 is the smallest number.
osboxes ... > Codes > c > ass_5 > timec ./3_smallest2
Wed 16 Mar 2022 01:16:48 AM IST

Enter the three numbers : 8 3 2
2 is the smallest number.
osboxes ... > Codes > c > ass_5 > timec ./3_smallest2
Wed 16 Mar 2022 01:17:06 AM IST
```

Solution 4

A screenshot of a Linux desktop environment. On the left, there is a file browser window showing a directory structure under 'Codes'. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'timec ./4_power'. The terminal output shows the program being run and its results. On the right, there is a calculator application.

```
osboxes ... > Codes > c > ass_5 > timec ./4_power
Wed 16 Mar 2022 01:17:39 AM IST

Enter a base, power : 5 3
Result (5^3) : 125
osboxes ... > Codes > c > ass_5 > timec ./4_power
Wed 16 Mar 2022 01:17:56 AM IST

Enter a base, power : 3 0
Result (3^0) : 1
osboxes ... > Codes > c > ass_5 > timec ./4_power
Wed 16 Mar 2022 01:18:06 AM IST

Enter a base, power : 0 3
Base should be greater than 0
osboxes ... > Codes > c > ass_5 > timec ./4_power
Wed 16 Mar 2022 01:18:15 AM IST

Enter a base, power : 7 -3
216
Exponent should greater than equal to 0
osboxes ... > Codes > c > ass_5 > timec ./4_power
Wed 16 Mar 2022 01:18:18 AM IST
```

Solution 5

A screenshot of a Linux desktop environment. In the top panel, there are several icons: assignment-Day5.pdf, Calculator, assign_2-5.docx - LibreOffice, ~/Desktop/Codes/c/ass_5/sumfnl.c (Codes) - Sublime Text (UNREGISTERED), and Linux Lite Terminal. The Sublime Text window shows a C program named 5_sumfnl.c. The terminal window shows the output of running the program, which calculates the sum of the first and last digits of a given number. The calculator window is also visible.

```
#include <stdio.h>
int main()
{
    int num;
    printf("Enter the number : ");
    scanf("%d", &num);
    int first_num, last_num = num%10, temp = num;
    while(num){
        first_num = num%10;
        num /= 10;
    }
    printf("\nSum of first and last number of %d: %d", temp);
    printf("\n-> %d + %d = %d \n", first_num, last_num, first_num+last_num);
    return 0;
}
```

Terminal Output:

```
osboxes ... > Codes > c > ass_5 > g_5_sumfnl 5_sumfnl.c
Wed 16 Mar 2022 01:22:34 AM IST
Enter the number : 3746839
Sum of first and last number of 3746839:
-> 3 + 9 = 12
osboxes ... > Codes > c > ass_5 >
```

Calculator Output:

```
Basic Mode
12x18 = 216
216
7 8 9 + - × ÷ C
4 5 6 × ( ) ÷
1 2 3 - x² √
0 . % + =
```

Solution 6

A screenshot of a Linux desktop environment. In the top panel, there are several icons: assignment-Day5.pdf, Calculator, assign_2-5.docx - LibreOffice, ~/Desktop/Codes/c/ass_5/6_output.c (Codes) - Sublime Text (UNREGISTERED), and Linux Lite Terminal. The Sublime Text window shows a C program named 6_output.c. The terminal window shows the output of running the program, which takes an integer input and prints its reverse. The calculator window is also visible.

```
#include <stdio.h>
int main()
{
    int n, x=0, y;
    printf("Enter an integer : ");
    scanf("%d", &n);
    int temp=n;
    //code to reverse number
    while(n!=0){
        y = n%10;           //gives last digit of n
        x = x*10+y;        //x storing reverse of input number
        n = n/10;           //remove one digit from end of n
    }
    printf("\nOutput is = %d [reverse of %d]\n", x, temp);
    return 0;
}
```

Terminal Output:

```
osboxes ... > Codes > c > ass_5 > g_6_output 6_output.c
osboxes ... > Codes > c > ass_5 > timec ./6_output
Wed 16 Mar 2022 01:31:30 AM IST
Enter an integer : 3452
Output is = 2543 [reverse of 3452]
osboxes ... > Codes > c > ass_5 >
```

Calculator Output:

```
Basic Mode
12x18 = 216
216
7 8 9 + - × ÷ C
4 5 6 × ( ) ÷
1 2 3 - x² √
0 . % + =
```

Solution 7

A screenshot of a Linux desktop environment. In the top right corner, there is a system tray icon for 'WR'. Below it is a terminal window titled 'Linux Lite Terminal' with the command 'timec ./7_output' and its output: 'True' and 'False'. To the left of the terminal is a code editor window titled 'ass_5' showing a C program. The code includes a conditional statement that prints 'True' or 'False' based on the value of 'a'. The terminal window also shows the command 'timec ./7_output' and its output: 'True' and 'False'. The desktop bar at the bottom has various icons for system functions like file manager, browser, and system settings.

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
    int a=1;
    if(a--) //a=1 true then a= 0
        printf("True\n"); //print true
    if(++a) //a=1 true
        printf("False\n"); //printf False
    return 0;
}
```

Solution 8

A screenshot of a Linux desktop environment. In the top right corner, there is a system tray icon for 'WR'. Below it is a terminal window titled 'Linux Lite Terminal' with the command 'timec ./8_output' and its output: '1500.00'. To the left of the terminal is a code editor window titled 'ass_5' showing a C program. The code includes a calculation where 'p' is set to 5000, 't' is 4, and 'r' is 7.5, then calculates 'si' as $(p*t*r)/100.0$. The terminal window also shows the command 'timec ./8_output' and its output: '1500.00'. The desktop bar at the bottom has various icons for system functions like file manager, browser, and system settings.

```
#include <stdio.h>
int main()
{
    int p,t;
    float r,si;
    p = 5000; t=4; r=7.5;
    si=(p*t*r)/100.0;
    printf("%.2f\n", si);
    return 0;
}
```

Solution 9

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons for system monitoring and user accounts. Below the panel, there is a dock with various application icons.

Sublime Text: A code editor window titled "9_output.c (Codes) - Sublime Text (UNREGISTERED)". It contains the following C code:

```
#include <stdio.h>
int main()
{
    int a=0101; //hex 0101 = dec 65
    printf("\n a=%d\n", a); //print decimal of a = 65
    return 0;
}
```

Terminal: A terminal window titled "Linux Lite Terminal". The command run was "g 9_output 9_output.c" followed by "timec ./9_output". The output shows:

```
a=65
```

Solution 10

The screenshot shows a Linux desktop environment with a dark theme, similar to the one above. The dock at the bottom includes icons for a file manager, terminal, and other applications.

Sublime Text: A code editor window titled "10_fnd_output.c (Codes) - Sublime Text (UNREGISTERED)". It contains the following C code:

```
#include <stdio.h>
int main()
{
    int var1=10, var2=6;
    if(var1==5) //var1 = 5
        var2++; //var2 = 7
    printf("%d %d\n", var1, var2++);
    return 0;
}
```

Terminal: A terminal window titled "Linux Lite Terminal". The command run was "g 10_fnd_output 10_fnd_output" followed by "timec ./10_fnd_output". The output shows:

```
5 7
```

Solution 11

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a directory structure under 'Codes'. In the center, there is a Sublime Text window displaying a C program named '11_foutput.c'. On the right, there is a terminal window titled 'Linux Lite Terminal' showing the output of a command to compile and run the program.

```
include <stdio.h>
int main()
{
    int i=0,j=1;
    printf("%d\n", i++&&j); // 0 && 1 = 0
    printf("%d %d\n", i, j); // i=1 j=1
    return 0;
}
```

```
osboxes ~ > Codes > c > ass_5 > g 11_foutput 11_foutput
Wed 16 Mar 2022 01:49:55 AM IST
0
1 1
osboxes ~ > Codes > c > ass_5 >
```

Solution 12

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a directory structure under 'New Folder'. In the center, there is a Sublime Text window displaying a C program named '12_count_digit.c'. On the right, there is a terminal window titled 'Linux Lite Terminal' showing the output of a command to compile and run the program.

```
// Write a C program to count total number of digits of an
// Integer number (n).
#include <stdio.h>
int main()
{
    int num;
    printf("Enter the number : ");
    scanf("%d", &num);

    int count=0, temp=num;
    do
    {
        count++;
    } while (num!=10);

    printf("the number %d contains %d digits.\n", temp, count);
    return 0;
}
```

```
osboxes ~ > Desktop > New Folder > ass_5 > g 12_count_digit 12_count_digit.c
osboxes ~ > Desktop > New Folder > ass_5 > ./12_count_digit
Enter the number : 345
the number 345 contains 3 digits.
osboxes ~ > Desktop > New Folder > ass_5 > ./12_count_digit
Enter the number : 12345
the number 12345 contains 5 digits.
osboxes ~ > Desktop > New Folder > ass_5 >
```

Solution 13

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for 'WB'. On the desktop, there is a 'New Folder' icon. In the bottom left, there are icons for 'pdf' and 'assignments'. The terminal window in the bottom right shows the command `./13_powof2` being run, which prompts for a number and outputs whether it is a power of 2 or not. The code editor window in the top left contains a C program named `13_powof2.c`.

```
1 // Write a C program to check whether the given number(n)
2 // can be expressed 512 as Power of Two (2) or not.
3
4 #include <stdio.h>
5
6 int main()
7 {
8     int num;
9
10    printf("Enter the number : ");
11    scanf("%d", &num);
12
13    int temp = num;
14
15    do num/=2;
16    while (num>=0);
17
18    if(num==1)
19        printf("%d is a number that can be expressed as power of 2.\n", temp);
20    else
21        printf("%d cannot be expressed as power of 2.\n", temp);
22
23    return 0;
24 }
```

```
osboxes ~ % cd Desktop/New Folder/ass_5/13_powof2.c
osboxes ~ % g 13_powof2 13_powof2.c
osboxes ~ % ./13_powof2
Enter the number : 512
512 is a number that can be expressed as power of 2.
osboxes ~ % cd Desktop/New Folder/ass_5
osboxes ~ % ./13_powof2
Enter the number : 84
84 cannot be expressed as power of 2.
osboxes ~ % cd Desktop/New Folder/ass_5
osboxes ~ % ./13_powof2
Enter the number : 64
64 is a number that can be expressed as power of 2.
osboxes ~ % cd Desktop/New Folder/ass_5
```

Solution 14

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for 'WB'. On the desktop, there is a 'New Folder' icon. In the bottom left, there are icons for 'pdf' and 'assignments'. The terminal window in the bottom right shows the command `./14_sumofseries` being run, which prompts for a number and calculates the sum of a series. The code editor window in the top left contains a C program named `14_sumofseries.c`.

```
1 // Write a C program to fin sum of following series
2 // where the value of n is taken as input:
3 // 1 + 1/2 + 1/3 + 1/4 + 1/5 + .. 1/n
4
5 #include <stdio.h>
6
7 int main()
8 {
9     int num; float sum = 0.0;
10
11    printf("Enter the number : ");
12    scanf("%d", &num);
13
14    for (int i = 1; i <= num; ++i)
15        sum = sum + (float)1/i;
16
17    printf("Sum of the series is : %.2f\n", sum);
18
19    return 0;
20 }
```

```
osboxes ~ % cd Desktop/New Folder/ass_5/14_sumofseries.c
osboxes ~ % g 14_sumofseries 14_sumofseries.c
osboxes ~ % ./14_sumofseries
Enter the number :
Sum of the series is : 2.28
osboxes ~ % cd Desktop/New Folder/ass_5
osboxes ~ % ./14_sumofseries
Enter the number : 8
Sum of the series is : 2.72
osboxes ~ % cd Desktop/New Folder/ass_5
osboxes ~ % ./14_sumofseries
Enter the number : 1
Sum of the series is : 1.00
osboxes ~ % cd Desktop/New Folder/ass_5
osboxes ~ % ./14_sumofseries
Enter the number : 2
Sum of the series is : 1.50
osboxes ~ % cd Desktop/New Folder/ass_5
```

Solution 15

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including LibreOffice, Network File Manager, Java, and a terminal window titled "Linux Lite Terminal". The terminal window has two tabs: "osboxes ~ > Desktop > New Folder > ass_5" and "osboxes ~ > Desktop > New Folder > ass_5". The terminal output for the first tab shows the execution of a C program named "15_pyramid" which prints a pyramid pattern of asterisks based on user input. The second tab shows the command being run again.

Code in 15_pyramid.c:

```
1 // Write a C program to print the following Pyramid pattern upto Nth row.
2 // Where N (number of rows to be printed) is taken as input.
3
4 #include <stdio.h>
5
6 int main()
7 {
8     int num;
9
10    printf("Enter the number : ");
11    scanf("%d", &num);
12
13    int t=num;
14    while(t){
15        while(t){
16            printf("* ");
17            t--;
18        }
19        printf("\n");
20        t = --num;
21    }
22
23    return 0;
24 }
```

Solution 16

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including LibreOffice, Network File Manager, Java, and a terminal window titled "Linux Lite Terminal". The terminal window has two tabs: "osboxes ~ > Desktop > New Folder > ass_5" and "osboxes ~ > Desktop > New Folder > ass_5". The terminal output for the first tab shows the execution of a C program named "16_perfect_number" which checks if a given number is a perfect number. The second tab shows the command being run again.

Code in 16_perfect_number.c:

```
1 // Write a C program to check whether a given number (n) is a perfect number or not?
2 // (Perfect Number - A perfect number is a positive integer number which is equals to
3 // the sum of its proper positive divisors.
4 // For example 6 is a perfect number because
5 // its proper divisors are 1, 2, 3 and it's sum is equals to 6.)
6
7 #include <stdio.h>
8
9 int main()
10 {
11     int num, sum=0;
12
13     printf("Enter the number : ");
14     scanf("%d", &num);
15
16     for (int i = 1; i < num; ++i){
17         if (num % i == 0)
18             sum += i;
19     }
20
21     if (sum == num)
22         printf("%d is a perfect number\n", num);
23     else
24         printf("%d is not a perfect number\n", num);
25
26     return 0;
27 }
```

Solution 17

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons and a clock showing 12:21:41 am. Below the panel, there are two windows:

- Sublime Text (UNREGISTERED)**: A code editor window titled "17_gcd.c" containing C code to find the Greatest Common Divisor (GCD) of two numbers using a for loop. The code includes comments explaining the logic.
- Linux Lite Terminal**: A terminal window showing command-line interactions. It runs the program "17_gcd" and prompts the user to enter two numbers. The terminal output shows multiple runs of the program with different inputs and outputs.

The desktop background is a colorful abstract image. At the bottom, there is a dock with various application icons.

```
1 // Write a program to find the GCD (Greatest Common
2 // Divisor) of 2 (two) numbers using 'for' loop.
3 // The two numbers are taken as input from the test cases.
4
5 #include <stdio.h>
6
7 int main()
8 {
9     int a, b, gcd=1;
10
11     printf("Enter the number : ");
12     scanf("%d %d", &a, &b);
13
14     int size = a>b?a:b;
15     for (int i = 1; i <= size; ++i){
16         if(a%i == 0 && b%i == 0)
17             gcd = i;
18     }
19     printf("GCD of the numbers %d and %d is %d\n", a, b, gcd);
20     return 0;
21 }
```

```
osboxes ~ > Desktop > New Folder > ass_5 g 17_gcd 17_gcd.c
osboxes ~ > Desktop > New Folder > ass_5 ./17_gcd
Enter the number : 6 12
GCD of the numbers 6 and 12 is 6
osboxes ~ > Desktop > New Folder > ass_5 ./17_gcd
Enter the number : 9 81
GCD of the numbers 9 and 81 is 9
osboxes ~ > Desktop > New Folder > ass_5 ./17_gcd
Enter the number : 4 12
GCD of the numbers 4 and 12 is 4
osboxes ~ > Desktop > New Folder > ass_5 ./17_gcd
Enter the number : 4 70
GCD of the numbers 4 and 70 is 2
osboxes ~ > Desktop > New Folder > ass_5
```

DAY 6

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a Java icon. The main window is Sublime Text, which is displaying a C program named `1_armstrong.c`. The code is designed to check if a given number is an Armstrong number. It includes comments explaining the logic: counting digits and calculating the sum of digits raised to the power of their count. The terminal window below shows the execution of the program for inputs 371 and 100, correctly identifying 371 as an Armstrong number and 100 as not being one.

```
1 // Write a Program to find if a given number is Armstrong number.
2 // Armstrong number has property:
3 // 153 = 1^3 + 5^3 + 3^3
4
5 #include <stdio.h>
6 #include <math.h>
7
8 int main(){
9     int num;
10
11     printf("Enter the number : ");
12     scanf("%d", &num);
13
14     int temp = num, arm=0, p=0;
15
16     //count number of digits in input number
17     while(num){
18         p++; num/=10;
19     }
20
21     //calculate armstrong number from input number
22     num = temp;
23     while(num){
24         arm += pow(num%10, p);
25         num/=10;
26     }
27
28     //print to tell number is armstrong or not
29     if(temp == arm)
30         printf("%d is an armstrong number.\n", temp);
31     else
32         printf("%d is not an armstrong number.\n", temp);
33 }
```

```
osboxes ~ > Desktop > New Folder > ass_6 ./1_armstrong
Enter the number : 371
371 is an armstrong number.
osboxes ~ > Desktop > New Folder > ass_6 ./1_armstrong
Enter the number : 100
100 is not an armstrong number.
osboxes ~ > Desktop > New Folder > ass_6 ./1_armstrong
Enter the number : 153
153 is an armstrong number.
osboxes ~ > Desktop > New Folder > ass_6
```

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a Java icon. The main window is Sublime Text, which is displaying a C program named `2_palindrome.c`. The code is designed to check if a given number is a palindrome. It uses a temporary variable to reverse the digits of the input number and compares it with the original number. The terminal window below shows the execution of the program for inputs 121 and 21, correctly identifying 121 as a palindrome and 21 as not being one.

```
1 // Write a program to find whether given number is palindrome or not.
2 // Palindrome number is a number which is same as its reverse
3
4 #include <stdio.h>
5
6 int main(int argc, char const *argv[])
7 {
8     int num, s=1, rem, pal=0, temp, p=1;
9
10    printf("Enter a number :");
11    scanf("%d", &num);
12
13    temp = num;
14
15    //checking place values
16    while(num>10){
17        p*=10;
18    }
19
20    num = temp;
21
22    //reversing the number
23    while(num){
24        rem = num%10;
25        pal += rem*p;
26        p/=10;
27        num/=10;
28    }
29
30    //comparing reverse number with original
31    if(pal == temp)
32        printf("%d is palindrome\n", pal);
33    else
34        printf("%d is not a palindrome\n", pal);
35 }
```

```
osboxes ~ > Desktop > New Folder > ass_6 g 2_palindrome 2_palindrome.c
osboxes ~ > Desktop > New Folder > ass_6 ./2_palindrome
Enter a number :121
121 is palindrome
osboxes ~ > Desktop > New Folder > ass_6 ./2_palindrome
Enter a number :21
21 is not a palindrome
osboxes ~ > Desktop > New Folder > ass_6 ./2_palindrome
Enter a number :11
11 is palindrome
osboxes ~ > Desktop > New Folder > ass_6 ./2_palindrome
Enter a number :4
4 is not a palindrome
osboxes ~ > Desktop > New Folder > ass_6
```

Solution 3

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a terminal icon. The main window is a Sublime Text editor titled "3_prime.c" which contains C code to check if a number is prime. Below it is a terminal window titled "Linux Lite Terminal" showing the execution of the program and its output for various test cases.

```
1 // Write a program in C to find whether a number n is prime number or not
2
3 #include <stdio.h>
4
5 int main(int argc, char const *argv[])
6 {
7     int num, c=0;
8
9     printf("Enter the number : ");
10    scanf("%d", &num);
11
12    for (int i = 1; i < num; ++i)
13    {
14        if(num|i == 0)
15            c++;
16
17        if(c==1 && num!=1)
18            printf("%d is prime\n", num);
19        else
20            printf("%d id not a prime\n", num);
21
22        return 0;
23    }
24
25 }
```

```
osboxes ~ > Desktop > New Folder > ass_6 > g 3_prime 3_prime.c
osboxes ~ > Desktop > New Folder > ass_6 > ./3_prime
Enter the number : 101
101 is prime
osboxes ~ > Desktop > New Folder > ass_6 > ./3_prime
Enter the number : 93
93 id not a prime
osboxes ~ > Desktop > New Folder > ass_6 > ./3_prime
Enter the number : 32
32 id not a prime
osboxes ~ > Desktop > New Folder > ass_6 > ./3_prime
Enter the number : 19
19 is prime
osboxes ~ > Desktop > New Folder > ass_6 >
```

Solution 4

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a terminal icon. The main window is a Sublime Text editor titled "4_prime10.c" which contains C code to find all prime numbers from 1 to n. Below it is a terminal window titled "Linux Lite Terminal" showing the execution of the program and its output for various test cases.

```
1 // Write a program in C to find all prime numbers from 1 to n
2
3 #include <stdio.h>
4
5 int checkPrime(int n){
6     int c=0;
7     for (int i = 2; i <= n; i++)
8         if(n|i == 0)
9             c++;
10
11    if(c == 1)  return 1;
12    else        return 0;
13
14 }
15
16 int main()
17 {
18     int num, c=0;
19
20     printf("Enter the number : ");
21     scanf("%d", &num);
22
23     for(int i=1; i<=num; i++){
24
25         if(checkPrime(i))
26             printf("%d ", i);
27
28         printf("\n");
29     }
30 }
```

```
osboxes ~ > Desktop > New Folder > ass_6 > g 4_prime10 4_prime10.c
osboxes ~ > Desktop > New Folder > ass_6 > ./4_prime10
Enter the number : 18
2 3 5 7 11 13 17
osboxes ~ > Desktop > New Folder > ass_6 > ./4_prime10
Enter the number : 10
2 3 5 7
osboxes ~ > Desktop > New Folder > ass_6 >
```

Solution 5

A screenshot of a Linux desktop environment. In the center is a Sublime Text window titled "5_find_output.c" containing C code. The code defines a function that finds the maximum value in an array. Below the Sublime Text window is a terminal window titled "Linux Lite Terminal" showing the command "g 5_find_output 5_find_output" and its output "12". The desktop background is a colorful gradient. Icons for "pdf" and "assignments" are visible on the desktop.

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
    int i, a[4]={3, 1, 12, 4}, result;
    if(result>a[i])
        continue;
    result=a[i];
}
//find max in array
for (int i = 0; i < 4; ++i)
{
    if(result>a[i])
        continue;
    result=a[i];
}
//print max value from array
printf("%d\n", result);
return 0;
```

```
osboxes ~ > Desktop > New Folder > ass_6 g 5_find_output 5_find_output
12
osboxes ~ > Desktop > New Folder > ass_6 ./5_find_output
```

Solution 6

A screenshot of a Linux desktop environment. In the center is a Sublime Text window titled "6_find_output.c" containing C code. The code defines a function that finds the third largest value in an array. Below the Sublime Text window is a terminal window titled "Linux Lite Terminal" showing the command "g 6_find_output 6_find_output" and its output "5". The desktop background is a colorful gradient. Icons for "pdf" and "assignments" are visible on the desktop.

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
    int arr[] = {1,2,3,4,5,6};
    int i, j, k;
    j = +arr[0]; //j = 4
    k = arr[1]++; //k = 2
    i = arr[j++]; //i = 5 and j = 5
    printf("i = %d, j = %d, k = %d\n", i, j, k);
    return 0;
}
```

```
osboxes ~ > Desktop > New Folder > ass_6 g 6_find_output 6_find_output
5
osboxes ~ > Desktop > New Folder > ass_6 ./6_find_output
```

Solution 7

A screenshot of a Linux desktop environment. In the center, there is a Sublime Text window titled "6/Desktop/New Folder/ass_6/7_find_output.c (New Folder) - Sublime Text (UNREGISTERED)". The code in the editor is:

```
#include <stdio.h>
int main()
{
    int a[6] = {1, 2, 3, 4, 5, 6};
    switch(sizeof(a)){
        case 1:
        case 2:
        case 3:
        case 4:
        case 5: printf("IIT KGP\n");
        break;
    }
    // print IIT MADRAS
    printf("IIT MADRAS \n");
    return 0;
}
```

To the right of the Sublime Text window is a terminal window titled "Linux Lite Terminal". The terminal shows the command:

```
osboxes ~ > Desktop > New Folder > ass_6 g 7_find_output 7_find_output
IIT MADRAS
osboxes ~ > Desktop > New Folder > ass_6
```

The desktop interface includes a dock at the bottom with various icons for file operations, system settings, and applications.

Solution 8

A screenshot of a Linux desktop environment. In the center, there is a Sublime Text window titled "6/Desktop/New Folder/ass_6/8_find_output.c (New Folder) - Sublime Text (UNREGISTERED)". The code in the editor is:

```
#include <stdio.h>
int main()
{
    int i = 0;
    char c = 'a';
    while(i < 5){
        i++;
        switch(c){
            case 'a': printf("%c\n", c); //5 times
            break;
        }
        printf("a\n"); //1 time
    }
    // -> total 6 times 'a' will print
    return 0;
}
```

To the right of the Sublime Text window is a terminal window titled "Linux Lite Terminal". The terminal shows the command:

```
osboxes ~ > Desktop > New Folder > ass_6 g 8_find_output 8_find_output
a
a
a
a
a
a
osboxes ~ > Desktop > New Folder > ass_6
```

The desktop interface includes a dock at the bottom with various icons for file operations, system settings, and applications.

Solution 9

The screenshot shows a Linux desktop environment with several windows open:

- Sublime Text (UNREGISTERED)**: A code editor displaying a C program named `9_find_output.c`. The code includes various array manipulation functions like `largelement`, `reverse`, `merge`, `dup`, `del`, `del_elem`, `sum`, `subtract`, `low_triangle`, `armstrong`, `palindrome`, `prime`, `prime10`, `prime100`, `find_output`, `5.find_output`, `6.find_output`, `7.find_output`, `8.find_output`, `9.find_output`, and `ass_7` through `ass_9`.
- Linux Lite Terminal**: A terminal window showing the command `g 9_find_output 9_find_output.c` being run, followed by the output of the program which is `4 4`.
- File Manager**: A window showing the directory structure: `Desktop/New Folder/ass_6` contains `g 9_find_output` and `9_find_output.c`.

At the bottom, the desktop dock displays icons for PDF files and assignments.

Solution 10

The screenshot shows a Linux desktop environment with a dark theme. A terminal window titled "Linux Lite Terminal -" is open, displaying the command-line interface. The code editor window, titled "10_largement_arr.c", contains C code for finding the largest element in an array. The code includes comments explaining the purpose of each section. The terminal window shows the execution of the program, reading input values (7, 60, 70, 200, 12, 40, -90, 60) and outputting the largest element as 200.

```
1 // Write a C Program to find Largest Element of an Integer Array.
2 // Here the number of elements in the array 'n' and the elements of
3 // the array is read from the test data.
4 // Use the printf statement given below to print the largest element.
5 // printf("Largest element = %d", largest);
6
7 #include <stdio.h>
8
9 int main()
10 {
11     int t;
12     scanf("%d", &t);
13
14     int a[t], max=-1000;
15     // printf("Enter the elements : ");
16
17     for (int i = 0; i < t; ++i)
18     {
19         scanf("%d", &a[i]);
20         if(max < a[i])
21             max = a[i];
22     }
23
24     printf("Largest element = %d\n", max);
25     return 0;
26 }
```

```
osboxes ~ > Desktop > New Folder > ass_6 g 10_largement_arr 10_largement_arr.c
osboxes ~ > Desktop > New Folder > ass_6 ./10_largement_arr
7
60
70
200
12
40
-90
60
Largest element = 200
osboxes ~ > Desktop > New Folder > ass_6
```

Solution 11

A screenshot of a Linux desktop environment. On the left, there is a file browser window showing a directory structure with files like ass_3, ass_4, ass_5, ass_6, ass_7, ass_8, ass_9, 11_reverse_arr.c, 12_merge_arr.c, 13_dup_del.c, 14_del_elem.c, 15_sum_ele.c, 16_subtract.c, 17_low_triangle.c, 18_armstrong.c, 19_armstrong.c, 20Palindrome.c, 21Palindrome.c, 23_prime.c, 24_prime10.c, 25_prime10.c, 26_prime10.c, 27_prime10.c, 28_prime10.c, 29_prime10.c, 30_prime10.c, 31_prime10.c, 32_prime10.c, 33_prime10.c, 34_prime10.c, 35_prime10.c, 36_prime10.c, and 37_prime10.c. A Sublime Text window is open with the file 11_reverse_arr.c. The code is as follows:

```
1 // Write a C Program to print the array elements in reverse order (Not
2 // reverse sorted order. Just the last element will become first element,
3 // second last element will become second element and so on)
4 // Here the size of the array, 'n' and the array elements is accepted from the
5 // test case data.
6
7 #include <stdio.h>
8
9 int main()
10 {
11     int t;
12     scanf("%d", &t);
13
14     int a[t];
15     printf("Enter the elements : ");
16
17     for (int i = 0; i < t; ++i){
18         scanf("%d", &a[i]);
19     }
20
21     //printing array in reserve order
22     printf("\nReverse elements : ");
23     while(t--){           // loop until t becomes 0
24         printf("%d ", a[t]);
25     }
26
27     printf("\n");
28     return 0;
29 }
```

The terminal window below shows the output of the program:

```
osboxes ~ > Desktop > New Folder > ass_6 g 11_reverse_arr 11_reverse_arr.c
osboxes ~ > Desktop > ass_6 ./11_reverse_arr
Enter the elements : 1 2 3 4 5 6
Reverse elements : 6 5 4 3 2 1
osboxes ~ > Desktop > New Folder > ass_6
```

Solution 12

A screenshot of a Linux desktop environment. On the left, there is a file browser window showing a directory structure with files like ass_3, ass_4, ass_5, ass_6, ass_7, ass_8, ass_9, 11_reverse_arr.c, 12_merge_arr.c, 13_dup_del.c, 14_del_elem.c, 15_sum_ele.c, 16_subtract.c, 17_low_triangle.c, 18_armstrong.c, 19_armstrong.c, 20Palindrome.c, 21Palindrome.c, 23_prime.c, 24_prime10.c, 25_prime10.c, 26_prime10.c, 27_prime10.c, 28_prime10.c, 29_prime10.c, 30_prime10.c, 31_prime10.c, 32_prime10.c, 33_prime10.c, 34_prime10.c, 35_prime10.c, 36_prime10.c, and 37_prime10.c. A Sublime Text window is open with the file 12_merge_arr.c. The code is as follows:

```
1 // C program to read Two One Dimensional Arrays of same data type
2 // (integer type) and merge them into another One Dimensional Array of same type.
3 // First take n as size of first array, then input all n elements of first array.
4 // Take m as size of second array, then input all m elements of second array.
5 // At last merge these two arrays to make third array of size n+m
6
7 #include <stdio.h>
8
9 int main(int argc, char const *argv[])
10 {
11     int n; scanf("%d", &n);
12
13     int a[n];
14     // printf("Enter the elements : ");
15     for (int i = 0; i < n; ++i)
16         scanf("%d", &a[i]);
17
18     int m; scanf("%d", &m);
19
20     int b[m];
21     // printf("Enter the elements : ");
22     for (int i = 0; i < m; ++i)
23         scanf("%d", &b[i]);
24
25     int s=n+m, c[s];
26     for (int i = 0; i < n; ++i)
27         c[i] = a[i];
28
29     for (int i = n, j=0; i < m, j<m; ++i, j++)
30         c[i] = b[j];
31
32     printf("\n");
33     for (int i= 0; i < s; ++i)
34         printf("%d\n", c[i]);
35
36     return 0;
37 }
```

The terminal window below shows the output of the program:

```
osboxes ~ > Desktop > New Folder > ass_6 g 12_merge_arr 12_merge_arr.c
osboxes ~ > Desktop > ass_6 ./12_merge_arr
3
15
45
25
3
60
70
80
15
45
25
60
70
80
osboxes ~ > Desktop > New Folder > ass_6
```

Solution 13

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a Java application. The main window is a Sublime Text editor with the title "-/Desktop/New Folder/ass_6/13_dup_del.c (New Folder) - Sublime Text (UNREGISTERED)". The code in the editor is:

```
1 // Write a C Program to delete duplicate elements from an array of integers.
2 // First take n as size of first array, then input all n elements of first array.
3 // Now delete duplicate elements from nested array. You can use nested
4 // loop for deleting duplicate elements.
5
6 #include <stdio.h>
7
8 int main()
9 {
10     int t; scanf("%d", &t);
11
12     int a[t];
13     for (int i = 0; i < t; ++i)
14         scanf("%d", &a[i]);
15
16     for (int i = 0; i < t; i++){
17         if(a[i]==a[i+1]) t--;
18         for (int j = i+1; j < t; j++)
19             if(a[i]==a[j]){
20                 for (int k = j; k < t-1; k++)
21                     a[k] = a[k+1];
22                 t--;
23             }
24     }
25     for (int i = 0; i < t; i++)
26         printf("%d", a[i]);
27
28     return 0;
29 }
```

The status bar at the bottom of the Sublime Text window indicates "Line 22, Column 29". Below the editor, there are two small icons: "pdf" and "assignments". To the right of the editor is a terminal window titled "Linux Lite Terminal" with the command:

```
osboxes ~ > Desktop > New Folder > ass_6 g 13_dup_del 13_dup_del.c
osboxes ~ > Desktop > New Folder > ass_6 ./13_dup_del
```

The terminal output shows the numbers 6, 6, 7, 2 on separate lines.

Solution 14

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a Java application. The main window is a Sublime Text editor with the title "-/Desktop/New Folder/ass_6/14_del_elem.c (New Folder) - Sublime Text (UNREGISTERED)". The code in the editor is:

```
1 // C Program to delete an element from a specified location of an Array starting
2 // from array [0] as the 1st position, array[1] as second position and so on.
3
4 #include <stdio.h>
5
6 int main()
7 {
8     int t; scanf("%d", &t);
9
10    int a[t];
11    for (int i = 0; i < t; ++i)
12        scanf("%d", &a[i]);
13
14    int pos; scanf("%d", &pos);
15
16    for (int i = pos; i < t; ++i)
17    {
18        int temp = a[i];
19        a[i] = a[i+1];
20        a[i+1] = temp;
21    }
22
23    printf("\n");
24    for (int i = 0; i < t-1; ++i)
25    {
26        printf("%d\n", a[i]);
27    }
28
29    return 0;
30 }
```

The status bar at the bottom of the Sublime Text window indicates "Line 15, Column 1". Below the editor, there are two small icons: "pdf" and "assignments". To the right of the editor is a terminal window titled "Linux Lite Terminal" with the command:

```
osboxes ~ > Desktop > New Folder > ass_6 g 14_del_elem 14_del_elem.c
osboxes ~ > Desktop > New Folder > ass_6 ./14_del_elem
```

The terminal output shows the numbers 5, 9, 10, 11, 20, 5, 9, 10, 11, 1 on separate lines.

Solution 15

A screenshot of a Linux desktop environment. In the foreground, there is a terminal window titled "Linux Lite Terminal" with the command `./15_sum_ele` running. The output shows the program's execution and the resulting sum of elements for a 3x3 matrix. In the background, there is a Sublime Text window showing a C program named "15_sum_ele.c". The code is a simple program that reads a 3x3 matrix from the user and prints the sum of each row. The terminal window shows the command being run and the resulting output.

```
1 // Write a C program to find the sum of all elements of each row of a matrix.
2 // Input number of rows and number of cols from user.
3 // Now input all rows*cols values from user
4 // Example: For a 3x3 matrix
5 // 4 5 6
6 // 6 7 3
7 // 1 2 3
8 // The output will be
9 // 15
10 // 16
11 // 6
12
13 #include <stdio.h>
14
15 int main(){
16     int arr[3][3], sum[3]={0, 0, 0};
17
18     for (int i = 0; i < 3; i++)
19         for (int j = 0; j < 3; j++)
20             scanf("%d", &arr[i][j]);
21
22     sum[i] += arr[i][j];
23
24     printf ("\n");
25     for (int i = 0; i < 3; i++)
26         printf ("%d\n", sum[i]);
27
28     return 0;
29 }
```

```
osboxes ~ > Desktop > New Folder > ass_6 g 15_sum_ele 15_sum_ele.c
4 5 6
6 7 3
1 2 3
15
16
6
osboxes ~ > Desktop > New Folder > ass_6
```

Solution 16

A screenshot of a Linux desktop environment. In the foreground, there is a terminal window titled "Linux Lite Terminal" with the command `./16_subtract` running. The output shows the program's execution and the resulting subtraction of two matrices. In the background, there is a Sublime Text window showing a C program named "16_subtract.c". The code is a program that subtracts matrix A from matrix B. It takes input for both matrices and prints the result. The terminal window shows the command being run and the resulting output.

```
1 // Write a C program to find subtraction of two matrices
2 // i.e. matrix A - matrix B=matrix C.
3 // Input number of rows and number of cols for both A and B matrix.
4 // Now input all rows*cols values from user
5 // If given matrix
6 // 2 3 5 1 5 2
7 // 4 5 6 2 3 4
8 // 6 5 7 3 3 4
9 // Output will be:
10 // 1 -2 3
11 // 2 2 2
12 // 3 2 3
13 // The elements of the output matrix are separated by one blank space
14
15 #include <stdio.h>
16
17 int main(){
18     int a[3][3], b[3][3], c[3][3];
19
20     for (int i = 0; i < 3; i++)
21         for (int j = 0; j < 3; j++)
22             scanf("%d", &a[i][j]);
23
24     printf("\n");
25     for (int i = 0; i < 3; i++)
26         for (int j = 0; j < 3; j++)
27             scanf("%d", &b[i][j]);
28
29     for (int i = 0; i < 3; i++)
30         for (int j = 0; j < 3; j++)
31             c[i][j] = a[i][j] - b[i][j];
32
33     printf("\n");
34     for (int i = 0; i < 3; i++){
35         for (int j = 0; j < 3; j++)
36             printf("%d ", c[i][j]);
37         printf("\n");
38     }
39
40 }
```

```
osboxes ~ > Desktop > New Folder > ass_6 g 16_subtract 16_subtract.c
2 3 5
4 5 6
6 5 7
1 5 2
2 3 4
3 3 4
1 -2 3
2 2 2
3 2 3
osboxes ~ > Desktop > New Folder > ass_6
```

Solution 17

The screenshot shows a Linux desktop environment with several windows open:

- A file manager window titled "final ass.docx - LibreOffice - ~/Desktop/New Folder..." is visible at the top.
- A terminal window titled "Linux Lite Terminal" is open, showing the command "g 17_low_triangle 17_low_triangle.c" and its output: 2 3 4, 5 6 7, 4 5 6.
- A Sublime Text window titled "17_low_triangle.c" is open, displaying C code for printing the lower triangle of a square matrix. The code includes comments explaining the logic and a main loop that prints elements from the matrix.
- The desktop background is a colorful gradient.
- Icons for "WB" and "New Folder" are visible on the desktop.

DAY 7

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. On the left, a Sublime Text window displays a C program named `1_count_UL.c`. The code counts uppercase and lowercase letters in a given string. On the right, a terminal window titled "Linux Lite Terminal" shows the output of running the program. The terminal window has tabs and shows the command `./1_count_UL` being run, with the output "Hello from CDAC hyderabad" and the counts "Uppercase Letters : 5" and "Lowercase Letters : 17".

```
1 // Write a C Program to Count Number of Uppercase and Lowercase Letters in a given string.
2 // The string may be a word or a sentence
3
4 #include <stdio.h>
5 #include <string.h>
6
7 int main()
8 {
9     int up_c=0, low_c=0;
10    char str[100]; // = "Hello from CDAC hyderabad";
11    fgets(str, 100, stdin);
12    for (int i = 0; i < strlen(str); ++i)
13    {
14        //check for lowercase
15        if(str[i] >= 97 && str[i] <= 122)
16            low_c++;
17        //check for uppercase
18        if(str[i] >= 65 && str[i] <= 90)
19            up_c++;
20    }
21    printf("\nUppercase Letters : %d\n", up_c);
22    printf("Lowercase Letters : %d\n", low_c);
23
24    return 0;
25 }
```

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. On the left, a Sublime Text window displays a C program named `2_word_length.c`. The code prints the largest and smallest words from a given sentence. On the right, a terminal window titled "Linux Lite Terminal" shows the output of running the program. The terminal window has tabs and shows the command `./2_word_length` being run, with the input "Enter the String : programming" and the output "Largest word is: programming" and "Smallest word is: C".

```
1 // Write a C program to print Largest and Smallest Word from a given sentence.
2 // If there are two or more words of same length then
3 // the first one is considered. Here single letter such as
4 // "I" , "a" is considered as a word.
5
6 #include<stdio.h>
7 #include<string.h>
8
9 int main()
10 {
11     char string[100] , arr[10][10];
12     int i = 0 , j = 0 , k = 0 , min , max;
13
14     printf("Enter the String : ");
15     scanf("%[^\\n]s" , string);
16
17     for(i = 0 ; i < strlen(string)+1 ; i++)
18     {
19         if(string[i] == ' ')
20             arr[k][j] = '\0';
21         j = 0;
22         k++;
23     }
24
25     arr[k][j] = '\0';
26
27     max = strlen(arr[0]);
28     min = strlen(arr[0]);
29
30     for(i = 0 ; i <= k ; i++)
31     {
32         if(max < strlen(arr[i]))
33             max = i;
34
35         if(min > strlen(arr[i]))
36             min = i;
37     }
38
39     printf("\nLargest word is: %s\nSmallest word is: %s
40
41     \n" , arr[max] , arr[min]);
42 }
```

Solution 3

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons and a clock showing 12:59:36 am. Below the panel, there are two windows: a Sublime Text editor and a terminal window.

Sublime Text Editor: The file being edited is `3_reverse.c`. The code is a C program that reads a string from the user and prints it in reverse. It includes comments explaining the purpose and how to run it.

```
1 // Write a C Program to reverse a given word using function.
2 // e.g. INDIA should be printed as AIDNI
3
4 #include <stdio.h>
5 #include <string.h>
6
7 int main()
8 {
9     char d[20];
10    printf("Enter the string : ");
11    scanf("%[^\\n]", d);
12
13    int l = strlen(d)-1;
14    while(l >= 0){
15        printf("%c", d[l]);
16        l--;
17    }
18    printf("\n");
19
20    return 0;
21 }
```

Terminal Window: The terminal window is titled "Linux Lite Terminal". It shows the command `./3_reverse` being run, followed by the input "INDIA" and the output "AIDNI". Another run of the program with the input "HERO" results in the output "OREH".

```
osboxes ~ > Desktop > New Folder > ass_7 g 3_reverse 3_reverse.c
osboxes ~ > Desktop > New Folder > ass_7 ./3_reverse
Enter the string : INDIA
The string after reverse : AIDNI
osboxes ~ > Desktop > New Folder > ass_7 ./3_reverse
Enter the string : HERO
The string after reverse : OREH
osboxes ~ > Desktop > New Folder > ass_7
```

Solution 4

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons and a clock showing 1:00:46 am. Below the panel, there are two windows: a Sublime Text editor and a terminal window.

Sublime Text Editor: The file being edited is `4_prime_check.c`. The code is a C program that checks if a given number is prime. It includes comments explaining the logic.

```
1 // Write a program to check if a given integer is Prime Number using function.
2 // Implement function: int checkPrime(int n);
3
4 #include <stdio.h>
5
6 int checkPrime(int n){
7     int c=0;
8     for (int i = 2; i <= n; i++)
9         if(n % == 0)
10             c++;
11
12     if(c == 1)  return 1;
13     else        return 0;
14 }
```

Terminal Window: The terminal window is titled "Linux Lite Terminal". It shows the command `./4_prime_check` being run, followed by the input "13" and the output "13 is prime number.". Another run with the input "11" results in the output "11 is prime number.". A run with the input "18" results in the output "18 is not a prime number.". A final run with the input "7" results in the output "7 is prime number.".

```
osboxes ~ > Desktop > New Folder > ass_7 g 4_prime_check 4_prime_check.c
osboxes ~ > Desktop > New Folder > ass_7 ./4_prime_check
Enter the number : 13
13 is prime number.
osboxes ~ > Desktop > New Folder > ass_7 ./4_prime_check
Enter the number : 11
11 is prime number.
osboxes ~ > Desktop > New Folder > ass_7 ./4_prime_check
Enter the number : 18
18 is not a prime number.
osboxes ~ > Desktop > New Folder > ass_7 ./4_prime_check
Enter the number : 7
7 is prime number.
osboxes ~ > Desktop > New Folder > ass_7
```

Solution 5

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a terminal. The main window is Sublime Text, which is displaying a C program named `5_power_rec.c`. The code is as follows:

```
1 // Write a C Program to find power of a given number using recursion.
2 // The number and the power to be calculated is taken from test case
3
4 #include <stdio.h>
5
6 int power(int num, int pw){
7     if(pw == 0)
8         return 1;
9     else
10        return num * power(num, pw-1);
11 }
12
13 int main()
14 {
15     int num, pw;
16     printf("Enter the number and power : ");
17     scanf("%d%d", &num, &pw);
18
19     printf("\n %d^%d: %d\n", num, pw, power(num, pw));
20
21     return 0;
22 }
```

The terminal window below shows the output of running the program. It prompts the user to enter a number and power, then prints the result.

```
osboxes ~ > Desktop > New Folder > ass_7 g 5_power_rec 5_power_rec.c
osboxes ~ > Desktop > New Folder > ass_7 ./5_power_rec
Enter the number and power : 5 3
5^3: 125
osboxes ~ > Desktop > New Folder > ass_7 ./5_power_rec
Enter the number and power : 16 3
16^3: 4096
osboxes ~ > Desktop > New Folder > ass_7
```

Solution 6

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a terminal. The main window is Sublime Text, which is displaying a C program named `6_binary.c`. The code is as follows:

```
1 // Write a C Program to print Binary Equivalent of an Integer using Recursion
2
3 #include <stdio.h>
4
5 int binary(int num){
6     if(num == 0)
7         return 0;
8     else
9         return num%2 + 10 * binary(num/2);
10 }
11
12 int main()
13 {
14     int num;
15     printf("Enter the number : ");
16     scanf("%d", &num);
17
18     printf("\n Binary(%d) : %d\n", num, binary(num));
19     return 0;
20 }
21
22
```

The terminal window below shows the output of running the program. It prompts the user to enter a number, then prints its binary representation.

```
osboxes ~ > Desktop > New Folder > ass_7 g 6_binary 6_binary.c
osboxes ~ > Desktop > New Folder > ass_7 ./6_binary
Enter the number : 30
Binary(30) : 11110
osboxes ~ > Desktop > New Folder > ass_7 ./6_binary
Enter the number : 10
Binary(10) : 1010
osboxes ~ > Desktop > New Folder > ass_7
```

Solution 7

A screenshot of a Linux desktop environment. On the left is a Sublime Text window titled "7/Desktop/New Folder/ass_7/7_find_output.c (New Folder) - Sublime Text (UNREGISTERED)". It contains the following C code:

```
#include <stdio.h>
int main()
{
    char str1[] = "Week-7-Assignment";
    char str2[] = {'W', 'e', 'e', 'k', '-', '7', '-', 'A', 's', 's', ' ', 'i', 'n', 'g', '\n', 'm', ' ', 'e', ' ', 'n', ' ', 't'};
    //str1 carries string which contain nullchar in the end 17
    //str2 is array of char ..size=16 of char here 17
    int n1 = sizeof(str1)/sizeof(str1[0]); // 18
    int n2 = sizeof(str2)/sizeof(str2[0]); // 17
    printf("n1 = %d, n2 = %d\n", n1, n2); //prints n1 = 18, n2 = 17
    return 0;
}
```

The terminal window at the bottom right is titled "Linux Lite Terminal". It shows the command being run and its output:

```
osboxes ~ > Desktop > New Folder > ass_7 g 7_find_output 7_find_output
osboxes ~ > Desktop > New Folder > ass_7 ./7_find_output
n1 = 18, n2 = 17
osboxes ~ > Desktop > New Folder > ass_7
```

Solution 8

A screenshot of a Linux desktop environment. On the left is a Sublime Text window titled "7/Desktop/New Folder/ass_7/8_find_output.c (New Folder) - Sublime Text (UNREGISTERED)". It contains the following C code:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str1[20] = "hello", str2[20] = " world";
    // strcat - concat both var string into str1
    // strcpy - copy 2nd arg string into str2
    strcpy(str1, str2);
    printf("%s\n", str1);
    //str1 = "hello world" and string copy to var str2 and then it prints "hello world"
    return 0;
}
```

The terminal window at the bottom right is titled "Linux Lite Terminal". It shows the command being run and its output:

```
osboxes ~ > Desktop > New Folder > ass_7 g 8_find_output 8_find_output
osboxes ~ > Desktop > New Folder > ass_7 ./8_find_output
hello world
osboxes ~ > Desktop > New Folder > ass_7
```

Solution 9

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several application icons and the system tray. Below the panel, there is a dock with various icons. A Sublime Text window is open, showing a file named `9_fill_blank.c`. The code contains a C program that prints a string "How is your exam". A terminal window titled "Linux Lite Terminal" is also open, showing the command `g 9_fill_blank 9_fill_blank.c` and its output: "Hwi orea".

```
#include <stdio.h>
int main()
{
    int i;
    char s[] = "How is your exam";
    for (int i = 0; s[i] != '\0'; ++i)
    {
        // to print "Hwi orea" the condition should be i%2==0
        if(i%2==0){
            printf("%c", s[i]);      //output : Hwi orea
        }
    }
    return 0;
}
```

Solution 10

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several application icons and the system tray. Below the panel, there is a dock with various icons. A Sublime Text window is open, showing a file named `10_value_i.c`. The code contains a C program that uses the `strcmp` function to compare two strings and prints the result. A terminal window titled "Linux Lite Terminal" is also open, showing the command `g 10_value_i 10_value_i.c` and its output: "0".

```
#include <stdio.h>
#include <string.h>
int main()
{
    static char str1[] = "dills";
    static char str2[20];
    static char str3[] = "daffo";
    int i;
    // strcmp function - string match then return 0 else return non-zero ..
    // i = 0 , string match
    i = strcmp(strcat(str3, strcpy(str2, str1)), "daffodills");
    // print 0
    printf("%d\n", i);
    return 0;
}
```

Solution 11

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a Java icon. The main window is a Sublime Text editor titled "11_find_output.c" located at "/Desktop/New Folder/ass_7/11_find_output.c". The code implements a function to reverse a character array or swap elements. The terminal window below it shows the command "g 11_find_output 11_find_output" being run, followed by the output "osboxes ~ > Desktop > New Folder > ass_7 g 11_find_output 11_find_output osboxes ~ > Desktop > New Folder > ass_7 ./11_find_output osboxes ~ > Desktop > New Folder > ass_7".

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

int main()
{
    char p[] = "assignment";
    char t;
    int i, j;

    // swapping of char array or reversing array
    for (int i = 0, j = strlen(p); i < j; i++)
    {
        t = p[i];
        p[i] = p[j-i];
        p[j-i] = t;
    }

    // '\0' or null char use to represent end of the character string.
    // during the reversing elements of array the '\0' null character comes at first
    // location p[0] = '\0' and '\0' also shows the termination of string
    // that is the reason the it prints nothing on output console
    printf("%s", p);
}

return 0;
```

Solution 12

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a Java icon. The main window is a Sublime Text editor titled "12_find_output.c" located at "/Desktop/New Folder/ass_7/12_find_output.c". The code defines a function "fu" that takes two integers and prints them. It then calls "fu" with arguments 67 and 67. The terminal window below it shows the command "g 12_find_output 12_find_output" being run, followed by the output "osboxes ~ > Desktop > New Folder > ass_7 g 12_find_output 12_find_output osboxes ~ > Desktop > New Folder > ass_7 ./12_find_output osboxes ~ > Desktop > New Folder > ass_7 67, 67 osboxes ~ > Desktop > New Folder > ass_7".

```
#include <stdio.h>
void fu(int x, int y);

int main(int argc, char const *argv[])
{
    char x=67, y='C'; // ASCII value of C = 67 assign to var y
    fu(x, y); // calling fu with passing values(67, 67)

    return 0;
}

void fu(int x, int y){
    printf("%d,%d\n", x, y); //print 67, 67
}
```

Solution 13

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a terminal icon. The main window is a Sublime Text editor titled "13_return_value.c" which contains the following C code:

```
#include <stdio.h>
int f(int n){
    static int i=1;
    if(n>=5)
        return i;
    else
        return f(n+1);
}
int main(){
    printf("%d\n", f(10)); // calling funciton f and print return value from f
    return 0;
}
```

To the right of the Sublime Text window is a terminal window titled "Linux Lite Terminal". The terminal shows the command "g 13_return_value 13_return_value" being run, followed by the output "osboxes ~ > Desktop > New Folder > ass_7 > g 13_return_value 13_return_value" and "10".

Solution 14

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a terminal icon. The main window is a Sublime Text editor titled "14_count_Hi.c" which contains the following C code:

```
#include <stdio.h>
int i; //i=0
int fun();
int main()
{
    while(i){ //0=false i.e while never start
        fun();
        main();
    }
    printf("Hello\n"); //print only Hello
    return 0;
}
int fun(){
    printf("Hi");
}
```

To the right of the Sublime Text window is a terminal window titled "Linux Lite Terminal". The terminal shows the command "g 14_count_Hi 14_count_Hi" being run, followed by the output "osboxes ~ > Desktop > New Folder > ass_7 > g 14_count_Hi 14_count_Hi" and "Hello".

Solution 15

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a terminal icon. The main window is a Sublime Text editor with a dark theme, displaying a C program named `15_find_output.c`. The code calculates the average of an array of ages. Below the editor is a dock with various application icons. To the right, a terminal window titled "Linux Lite Terminal" shows the command `g 15_find_output 15_find_output` being run, followed by the output `27.08`.

```
#include <stdio.h>
float func(float age[]);
int main()
{
    float result, age[]={23.4, 55, 22.6, 3, 40.5, 18};
    result = func(age); //calling func function assign return value to result
    printf("%.2f\n", result); //print result avg of age = 27.08
    return 0;
}
float func(float age[]){
    int i;
    float result, sum=0.0;
    for (int i = 0; i < 6; ++i){ //count sum of age
        sum += age[i];
    }
    result = (sum/6); //result = avg of age
    return result;
}
```

Solution 16

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a terminal icon. The main window is a Sublime Text editor with a dark theme, displaying a C program named `16_find_output.c`. The code uses a loop to print the number 10 five times. Below the editor is a dock with various application icons. To the right, a terminal window titled "Linux Lite Terminal" shows the command `g 16_find_output 16_find_output` being run, followed by the output `10` repeated five times.

```
#include <stdio.h>
int main()
{
    int i;
    for (int i = 0; i < 5; i++) //loop 5 times
    {
        int i=10; //assign i=10 in every loop
        printf("%d\n", i); //print 10
        i++; //10+1
    }
    return 0;
}
// print number 10 five times because before printing variable, 10 is assigning to
// variable i in every loop
```

Solution 17

A screenshot of a Linux desktop environment. On the left, a Sublime Text window displays a C program named `17_count_factexc.c`. The code calculates the factorial of 10 using recursion. On the right, a terminal window titled "Linux Lite Terminal" shows the command `./17_count_factexc` being run, which outputs the value `3628800`.

```
#include <stdio.h>
int factorial(int);
int main()
{
    int n=10;
    long int f;
    f = factorial(n); //function calling
    printf("%d = %ld\n", n, f);
    return 0;
}
// factorial func is called only one times which means fuctions executed one times only.
int factorial(int n){
    if(n==0)
        return 1;
    else
        return n*factorial(n-1);
}
```

```
osboxes ~ > Desktop > New Folder > ass_7 g 17_count_factexc 17_count_factexc.c
osboxes ~ > Desktop > New Folder > ass_7 ./17_count_factexc
10! = 3628800
osboxes ~ > Desktop > New Folder > ass_7
```

Solution 18

A screenshot of a Linux desktop environment. On the left, a Sublime Text window displays a C program named `18_find_output.c`. It contains a swap function and a main function that swaps the values of `num1` and `num2`. On the right, a terminal window titled "Linux Lite Terminal" shows the command `./18_find_output` being run, demonstrating that the swap did not occur because of pass-by-value.

```
#include <stdio.h>
void swap(int a, int b){
    int temp;
    temp = a;
    a = b;
    b = temp;
}
int main()
{
    int num1 = 10, num2 = 20;
    printf("Before swapping num1 = %d num2 = %d\n", num1, num2);
    swap(num1, num2);
    printf("After swapping num1 = %d num2 = %d\n", num1, num2);
    return 0;
}
// output
// Before swapping num1 = 10 num2 = 20
// After swapping num1 = 10 num2 = 20
// swapping not happened because here value passing with call by value not using call by reference.
// Due to that the changes made in a function is not reflected in real values
```

```
osboxes ~ > Desktop > New Folder > ass_7 g 18_find_output 18_find_output.c
osboxes ~ > Desktop > New Folder > ass_7 ./18_find_output
Before swapping num1 = 10 num2 = 20
After swapping num1 = 10 num2 = 20
osboxes ~ > Desktop > New Folder > ass_7
```

Solution 19

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons and a clock showing 7:52:59 am. Below the panel, there are two windows:

- Sublime Text (UNREGISTERED)**: A code editor window titled "19_find.c". It contains C code for finding the middle value between three parameters. The code includes a recursive function "func" and a main function that prints the result. The file path is "/Desktop/New Folder/ass_7/19_find.c".
- Linux Lite Terminal**: A terminal window with the title "Linux Lite Terminal -". It shows the command "g 19_find 19_find.c" being run, followed by the output "21". The terminal path is "osboxes ~ > Desktop > New Folder > ass_7".

The desktop background is a colorful gradient. There are some icons on the desktop, including "WB" and "New Folder".

```
1 // The function func() is used to find...
2
3 #include <stdio.h>
4
5 //function is use to find middle value between 3 parameteric values
6 int func(int a, int b, int c){
7     if (a==b && c==b)
8         return b;
9     else if(a>b)
10        return func(a, c, b);
11    else
12        return func(b, a, c);
13 }
14
15 int main()
16 {
17     printf("%d\n", func(21, 31, 15)); // middle value is 21
18     return 0;
19 }
```

```
osboxes ~ > Desktop > New Folder > ass_7 g 19_find 19_find.c
21
osboxes ~ > Desktop > New Folder > ass_7
```

DAY 8

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. On the left is a Sublime Text window titled "1_multi.c" containing C code for multiplication using recursion. On the right is a terminal window titled "Linux Lite Terminal" showing the output of running the program.

```
1 // Write a program to implement multiplication using addition. Use recursion.
2
3 #include <stdio.h>
4
5 int multi(int a, int b){ //3 2
6     static int sum = 0;
7
8     if(b == 0)
9         return sum;
10    else
11        sum += a;
12        multi(a, b-1);
13    }
14
15 int main(){
16     int num1, num2;
17
18     printf("Enter the number : ");
19     scanf("%d", &num1, &num2);
20
21     printf("%d * %d = %d\n", num1, num2, multi(num1, num2));
22
23     return 0;
24 }
```

```
osboxes: ~ > Desktop > New Folder > ass_8 > g 1_multi 1_multi.c
osboxes: ~ > Desktop > New Folder > ass_8 > ./1_multi
Enter the number : 5 3
5 * 3 = 15
osboxes: ~ > Desktop > New Folder > ass_8
```

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. On the left is a Sublime Text window titled "1_multi.c" containing C code for swapping two numbers. On the right is a terminal window titled "Linux Lite Terminal" showing the output of running the program.

```
1 // Write a program to swap two numbers.
2
3 #include <stdio.h>
4
5 int swap(int *a, int *b){
6
7     int t = *a;
8     *a = *b;
9     *b = t;
10 }
11
12 int main(){
13
14     int a, b;
15     printf("Enter the two numbers : ");
16     scanf("%d %d", &a, &b);
17
18     printf("Before swapping : a = %d, b = %d \n", a, b);
19     swap(&a, &b);
20     printf("After swapping : a = %d, b = %d \n", a, b);
21
22 }
```

```
osboxes: ~ > Desktop > New Folder > ass_8 > g 1_multi 1_multi.c
osboxes: ~ > Desktop > New Folder > ass_8 > ./1_multi
Enter the number : 5 3
5 * 3 = 15
osboxes: ~ > Desktop > New Folder > ass_8 > g 2_swap 2_swap.c
osboxes: ~ > Desktop > New Folder > ass_8 > ./2_swap
Enter the two numbers : 6 9
Before swapping : a = 6, b = 9
After swapping : a = 9, b = 6
osboxes: ~ > Desktop > New Folder > ass_8
```

Solution 3

The screenshot shows a Linux desktop environment with a dark theme. On the left is a Sublime Text window titled `~/Desktop/New Folder/ass_8/3_sum.c (New Folder) - Sublime Text (UNREGISTERED)`. It contains C code for a recursive function to find the sum of integers. On the right is a terminal window titled `Linux Lite Terminal` showing the output of running the program.

```
1 // Write a recursive function to find the sum of n integers.
2
3 #include <stdio.h>
4
5 int sum(int num){
6     static int s = 0;
7     if(num==0)
8         return s;
9     else
10    {
11        s += num;
12        sum(num-1);
13    }
14 }
15
16 int main(){
17     int num;
18
19     printf("Enter the number : ");
20     scanf("%d", &num);
21
22     printf("Sum of integers : %d\n", sum(num));
23     return 0;
24 }
```

```
osboxes ~ Desktop > New Folder > ass_8 g 3_sum 3_sum.c
osboxes ~ Desktop > New Folder > ass_8 ./3_sum
Enter the number : 5
Sum of integers : 15
osboxes ~ Desktop > New Folder > ass_8
```

Solution 4

The screenshot shows a Linux desktop environment with a dark theme. On the left is a Sublime Text window titled `~/Desktop/New Folder/ass_8/4_concat.c (New Folder) - Sublime Text (UNREGISTERED)`. It contains C code for concatenating two strings without using library functions. On the right is a terminal window titled `Linux Lite Terminal` showing the output of running the program.

```
1 // Write a c program to concatenate two strings without using library functions
2
3 #include <stdio.h>
4
5 int main(){
6     char a[] = "Hello", b[] = " world";
7     char s[40];
8
9     for (int i = 0, j = 0, k = 0; j<sizeof(b); ++i)
10    {
11        if(a[i] == '\0')
12            continue;
13
14        if(i < sizeof(a))
15            s[k] = a[i], k++;
16
17        else
18            s[k] = b[j], j++, k++;
19
20    }
21
22
23    printf("%s\n", s);
24
25    return 0;
26 }
```

```
osboxes ~ Desktop > New Folder > ass_8 g 4_concat 4_concat.c
osboxes ~ Desktop > New Folder > ass_8 ./4_concat
Hello world
osboxes ~ Desktop > New Folder > ass_8
```

Solution 5

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a directory structure with files like ass_5, ass_6, ass_7, ass_8, ass_9, j.java, and pow.c. In the center, there is a Sublime Text window with multiple tabs open, showing code for a C program to copy strings without using library functions. The active tab contains the following code:

```
1 // Write a c program to copy string without using library functions
2
3 #include <stdio.h>
4
5 void copy(char a[], char s[], int n){
6     for (int i = 0; i < n; ++i)
7         s[i] = a[i];
8 }
9
10 int main(){
11     char a[] = "Welcome to Future", s[10];
12
13     copy(a, s, sizeof(a));
14
15     printf("%s\n", s);
16
17     return 0;
18 }
```

On the right, there is a terminal window titled "Linux Lite Terminal" showing the command line and its output. The command entered was "g 5_copy_string 5_copy_string.c ./5_copy_string". The output shows the string "Welcome to Future" being printed.

Solution 6

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a directory structure with files like ass_5, ass_6, ass_7, ass_8, ass_9, j.java, and pow.c. In the center, there is a Sublime Text window with multiple tabs open, showing code for a C program to find the length of a string without using library functions. The active tab contains the following code:

```
1 // Write a c program to find length of string without using library functions
2
3 #include <stdio.h>
4
5 int main(){
6     char a[] = "Programmer";
7     int size=0;
8
9     for (int i = 0; a[i] != '\0' ; ++i)
10         size++;
11
12     printf("String : %s\n", a);
13     printf("size of string : %d\n", size);
14
15     return 0;
16 }
```

On the right, there is a terminal window titled "Linux Lite Terminal" showing the command line and its output. The command entered was "g 6_string_len 6_string_len.c ./6_string_len". The output shows the string "Programmer" and its size, which is 10.

DAY 9

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a terminal window titled "Linux Lite Terminal". The terminal window displays the command-line interface of a Linux system. In the bottom panel, there is a dock with various application icons.

Sublime Text Window:

```
1 // Write a C program to calculate power a given number using pow(a,b) function in math.h with evaluates: a^b
2 // Use #include<math.h> in your c program.
3
4 #include <stdio.h>
5 #include <math.h>
6
7 int power(int number, int pow){
8     if(pow == 1)
9         return number;
10    else
11        return number * power(number, pow-1);
12 }
13
14 int main()
15 {
16     double num, pw;
17
18     printf("Enter the number : ");
19     scanf("%lf%lf", &num, &pw);
20
21     printf("%.2lf\n", pow(num, pw));
22     return 0;
23 }
```

Terminal Window:

```
osboxes ~ Desktop > New Folder > ass_9 > cd ../ass_9
osboxes ~ Desktop > New Folder > ass_9 > g 1_pow 1_pow.c
/usr/bin/ld: /tmp/cc4ntExw.o: in function `main':
1_pow.c(.text+0xb): undefined reference to `pow'
collect2: error: ld returned 1 exit status
osboxes ~ Desktop > New Folder > ass_9 > 1.1_pow
osboxes ~ Desktop > New Folder > ass_9 > ./1_pow
Enter the number : 2 5
32.000000
osboxes ~ Desktop > New Folder > ass_9 > g 1_pow 1_pow.c -lm
osboxes ~ Desktop > New Folder > ass_9 > ./1_pow
Enter the number : 5 2
25.00
osboxes ~ Desktop > New Folder > ass_9 > ./1_pow
Enter the number : 3 0.5
1.73
osboxes ~ Desktop > New Folder > ass_9 >
```

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a terminal window titled "Linux Lite Terminal". The terminal window displays the command-line interface of a Linux system. In the bottom panel, there is a dock with various application icons.

Sublime Text Window:

```
1 // Write a program and implement a function to find if it is leap year or not.
2 // * is_leap_year() returns 1 if leap year
3 // * !is_leap_year() returns 0 if NOT leap year
4 // Take 4 digit year as input for example: 2000
5
6 #include <stdio.h>
7
8 int is_leap_year(int year){
9     if(year%400==0 || year%100 != 0 && year%4==0)
10        return 1;
11    else
12        return 0;
13 }
14
15 int main(){
16     int year;
17
18     printf("Enter the year : ");
19     scanf("%d", &year);
20
21     if(is_leap_year(year))
22         printf("\n%d is a leap year \n", year);
23     else
24         printf("\n%d is not a leap year \n", year);
25     return 0;
26 }
```

Terminal Window:

```
osboxes ~ Desktop > New Folder > ass_9 > g 2_leap_yr 2_leap_yr.c
osboxes ~ Desktop > New Folder > ass_9 > ./2_leap_yr
Enter the year : 2000
2000 is a leap year
osboxes ~ Desktop > New Folder > ass_9 > ./2_leap_yr
Enter the year : 2012
2012 is a leap year
osboxes ~ Desktop > New Folder > ass_9 > ./2_leap_yr
Enter the year : 2022
2022 is not a leap year
osboxes ~ Desktop > New Folder > ass_9 >
```

Solution 3

A screenshot of a Linux desktop environment. In the foreground, there is a Sublime Text window titled "3_fact.c" containing C code for calculating factorial. Below it is a terminal window titled "Linux Lite Terminal" showing the execution of the program and its output for factorial(5) and factorial(4). The desktop background is a colorful gradient, and the taskbar at the bottom shows various application icons.

```
3_fact.c
1 // Write a program and implement a factorial function using below
2 // prototype declarations: long int factorial(long int);
3
4 #include <stdio.h>
5 long int factorial(long int);
6
7 int main(){
8     long int num;
9     printf("Enter the number : ");
10    scanf("%ld", &num);
11
12    printf("factorial(%ld) : %ld\n", num, factorial(num));
13
14    return 0;
15 }
16
17 long int factorial(long int num){
18     if(num==1)
19         return 1;
20     else
21         return num * factorial(num-1);
22 }
```

```
Linux Lite Terminal -
osboxes: ~/Desktop/New Folder/ass_9/3_fact.c
osboxes: ~/Desktop/New Folder/ass_9/3_fact ./3_fact
Enter the number : 5
factorial(5) : 120
osboxes: ~/Desktop/New Folder/ass_9/3_fact
Enter the number : 4
factorial(4) : 24
osboxes: ~/Desktop/New Folder/ass_9/3_fact
```

Solution 4

A screenshot of a Linux desktop environment. In the foreground, there is a Sublime Text window titled "4_avg.c" containing C code for calculating the average of two numbers. Below it is a terminal window titled "Linux Lite Terminal" showing the execution of the program and its output for various inputs. The desktop background is a colorful gradient, and the taskbar at the bottom shows various application icons.

```
4_avg.c
1 // Implement average function of two numbers as below:
2 // double average(double, double);
3
4 #include <stdio.h>
5
6 double average(double num1, double num2){
7     return (num1+num2)/2;
8 }
9
10 int main(){
11     double num1, num2;
12
13     printf("Enter two numbers : ");
14     scanf("%lf%lf", &num1, &num2);
15
16     printf("\nAverage(%lf, %lf) : %.2lf\n", num1, num2, average(num1, num2));
17
18     return 0;
19 }
```

```
Linux Lite Terminal -
osboxes: ~/Desktop/New Folder/ass_9/4_avg.c
osboxes: ~/Desktop/New Folder/ass_9/4_avg ./4_avg
Enter two numbers : 2 8.4
Average(2.0, 8.4) : 5.20
osboxes: ~/Desktop/New Folder/ass_9/4_avg
Enter two numbers : 2.1 3.4
Average(2.1, 3.4) : 2.75
osboxes: ~/Desktop/New Folder/ass_9/4_avg
Enter two numbers : 30 20
Average(30.0, 20.0) : 25.00
osboxes: ~/Desktop/New Folder/ass_9/4_avg
```

Solution 5

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a terminal window titled "Linux Lite Terminal". The terminal window displays the command "osboxes ~ > Desktop > New Folder > ass_9 g 5_exp 5_exp.c" followed by the output of the program. The main workspace contains two windows: one for Sublime Text showing the code for "5_exp.c" and another for a file manager showing a folder structure.

```
5_exp.c
1 // Write a program to use exp function from math.h
2
3 #include <stdio.h>
4 #include <math.h>
5
6 int main(){
7     double num;
8
9     printf("Enter the number : ");
10    scanf("%lf", &num);
11
12    printf("\nexp (%.2lf) : %.4lf\n", num, exp(num));
13
14    return 0;
15 }
```

```
osboxes ~ > Desktop > New Folder > ass_9 g 5_exp 5_exp.c
/usr/bin/ld: /tmp/ccjdixFy.o: in function `main':
5_exp.c(.text+0x4e): undefined reference to `exp'
collect2: error: ld returned 1 exit status
osboxes ~ > Desktop > New Folder > ass_9 1 g 5_exp 5_exp.c -lm
osboxes ~ > Desktop > New Folder > ass_9 ./5_exp
Enter the number : 2.2
exp (2.20) : 9.0250
osboxes ~ > Desktop > New Folder > ass_9
```

Solution 6

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a terminal window titled "Linux Lite Terminal". The terminal window displays the command "osboxes ~ > Desktop > New Folder > ass_9 g 6_arr_loc 6_arr_loc.c" followed by the output of the program. The main workspace contains two windows: one for Sublime Text showing the code for "6_arr_loc.c" and another for a file manager showing a folder structure.

```
6_arr_loc.c
1 // Write a program to print all the locations at which a particular element (taken as input) is found in a array
2 // and also print the total number of times it occurs in the array. The location starts from 1.
3 // For example if there are 4 elements in the array
4 // 5
5 // 6
6 // 5
7 // 7
8 // If the element to search is 5 then the output will be
9 // 5 is present at location 1
10 // 5 is present at location 3
11
12 #include <stdio.h>
13
14 int main(){
15     int t; scanf("%d", &t);
16     int arr[t];
17
18     for (int i = 0; i < t; i++)
19         scanf("%d", &arr[i]);
20
21     int search, count=0;
22     scanf("%d", &search);
23
24     for (int i = 0; i < t; i++)
25         if(search == arr[i])
26             printf("%d is present at location : %d\n", search, i+1);
27             count++;
28
29
30     if(count == 0)
31         printf("%d is not present in array \n", search);
32     else
33         printf("%d is present %d times in an array\n", search, count);
34
35
36     return 0;
37 }
38
```

```
osboxes ~ > Desktop > New Folder > ass_9 g 6_arr_loc 6_arr_loc.c
osboxes ~ > Desktop > New Folder > ass_9 ./6_arr_loc
7
30
50
90
30
70
50
30
30
30
30
30 is present at location : 1
30 is present at location : 4
30 is present at location : 6
30 is present at location : 7
30 is present 4 times in an array
osboxes ~ > Desktop > New Folder > ass_9 ./6_arr_loc
4
50
60
20
10
80
80 is not present in array
osboxes ~ > Desktop > New Folder > ass_9
```

Solution 7

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons for system monitoring and user accounts. Below the panel, there is a dock with various application icons.

Sublime Text: A file named `7_L_search.c` is open. The code implements a linear search function to find a target value in an array. It includes `#include <stdio.h>` and `#include <stdlib.h>`. The main loop iterates through the array, comparing each element with the search value. If a match is found, it prints the index and exits. If no match is found, it prints a message indicating the value is not present.

```
7_L_search.c
1 // Write a C program to search a given element from a 1D array and display
2 // the position at which it is found by using linear search function. The index location starts from 1.
3
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 int main(){
8     int t; scanf("%d", &t);
9     int arr[t];
10    for (int i = 0; i < t; i++)
11        scanf("%d", &arr[i]);
12
13    int search;
14    scanf("%d", &search);
15
16    for (int i = 0; i < t; i++)
17        if(search == arr[i])
18            printf("\n%d is present at location : %d\n", search, i+1), exit(0);
19
20    printf("\n%d is not present in the array\n", search);
21
22    return 0;
23 }
```

Terminal: The terminal window is titled "Linux Lite Terminal". It shows the command `g 7_L_search 7_L_search.c` being run, followed by the output of the program. The output shows the array elements 4, 45, 65, 85, 25, and 95. It then asks for a search value, receives 6, and outputs "6 is present at location : 1".

```
osboxes ~ > Desktop > New Folder > ass_9 g 7_L_search 7_L_search.c
4
45
65
85
25
95

95 is not present in the array
osboxes ~ > Desktop > New Folder > ass_9 .7_L_search
5
6
9
5
4
4
7
6

6 is present at location : 1
osboxes ~ > Desktop > New Folder > ass_9
```

Solution 8

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons for system monitoring and user accounts. Below the panel, there is a dock with various application icons.

Sublime Text: A file named `8_revse_arr.c` is open. The code reverses an array by creating a new array and copying elements from the end of the original array to the new array. It includes `#include <stdio.h>`. The program first prints the original array, then prints the reversed array.

```
8_revse_arr.c
1 // Write a C program to reverse an array by using another new array
2
3 #include <stdio.h>
4
5 int main(){
6     int t; scanf("%d", &t);
7     int arr[t], rev_arr[t];
8
9     for (int i = 0; i < t; i++)
10        scanf("%d", &arr[i]);
11
12    for (int i = 0, k = t-1; i < t; i++, k--)
13        rev_arr[i] = arr[k];
14
15    printf("\nOriginal array elements are : \n");
16    for (int i = 0; i < t; i++)
17        printf("%d ", arr[i]);
18
19    printf("\n\nReversed array elements are : \n");
20    for (int i = 0; i < t; i++)
21        printf("%d ", rev_arr[i]);
22
23    printf("\n");
24    return 0;
25 }
```

Terminal: The terminal window is titled "Linux Lite Terminal". It shows the command `g 8_revse_arr 8_revse_arr.c` being run, followed by the output of the program. The output shows the original array elements 8, 9, 10, 6, 4, 7, 11, and the reversed array elements 11, 7, 4, 6, 10, 9, 8.

```
osboxes ~ > Desktop > New Folder > ass_9 g 8_revse_arr 8_revse_arr.c
8
9
10
6
4
7
11

Original array elements are :
8 9 10 6 4 7 11

Reversed array elements are :
11 7 4 6 10 9 8
osboxes ~ > Desktop > New Folder > ass_9 ./8_revse_arr
```

Solution 9

The screenshot shows a Linux desktop environment with several windows open:

- File Manager:** Shows a folder structure with files like 8.reverse_arr.c, 9.swap_arr.c, and various .c and .h files.
- Sublime Text (UNREGISTERED):** An editor window containing the code for 9_swap_arr.c. The code is a C program that reverses an array by swapping elements without using a new array. It includes comments explaining the logic and uses standard C libraries.
- Terminal:** A terminal window titled "Linux Lite Terminal" showing the command ./9_swap_arr and its output. The output shows the original array [50, 60, 40, 30] and the reversed array [20, 30, 40, 60].

```
1 // Write a C program to reverse an array by swapping the elements and without using any new array
2
3 #include <stdio.h>
4
5 int main(){
6     int t; scanf("%d", &t);
7     int arr[t];
8
9     for (int i = 0; i < t; i++)
10        scanf("%d", &arr[i]);
11
12    for (int i = 0, k = t-1; i < t/2; ++i, -k) {
13        int t = arr[i];
14        arr[i] = arr[k];
15        arr[k] = t;
16    }
17    printf("\nReversed array elements are : \n");
18    for (int i = 0; i < t; i++)
19        printf("%d ", arr[i]);
20
21    printf("\n\n");
22
23 }
```

Line 21, Column 13 - Field 1 of 3

Spaces: 4 C

DAY 10

Structures and Unions

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are icons for LibreOffice, a file folder, assignments, and a terminal window. The terminal window is titled "Linux Lite Terminal" and shows the command line and output of a C program named "1_std_detail.c". The code defines a structure "std" with fields id, gender, and name, and a main function that reads input for these fields and prints them. The output shows the student ID, gender, and name being entered and printed.

```
1_std_detail.c
1 // Write a simple structure called student containing fields (int id, char gender and char name[size]); Create a
2 // variable of this structure(ex: std1), fill in the fields of this structure variable
3
4 #include <stdio.h>
5
6 struct std{
7     int id;
8     char gender;
9     char name[20];
10 }
11
12 int main(){
13     struct std std1;
14
15     printf("Enter the student ID : ");
16     scanf("%d", &std1.id);
17
18     printf("Enter student gender : ");
19     scanf(" %c", &std1.gender);
20
21     printf("Enter student name : ");
22     scanf(" %[^\\n]", std1.name);
23
24     printf("\nStudent ID      : %d\n", std1.id);
25     printf("Student gender : %c\n", std1.gender);
26     printf("Student name   : %s\n", std1.name);
27
28     return 0;
29 }
```

```
osboxes ~ > Desktop > New Folder > ass_10 > struct > g 1_std_detail 1_std_detail.c
osboxes ~ > New Folder > ass_10 > struct > ./1_std_detail
Enter the student ID : 101
Enter student gender : M
Enter student name : Itachi
Student ID      : 101
Student gender : M
Student name   : Itachi
osboxes ~ > New Folder > ass_10 > struct
```

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are icons for LibreOffice, a file folder, assignments, and a terminal window. The terminal window is titled "Linux Lite Terminal" and shows the command line and output of a C program named "2_personal.c". The code defines a structure "personal" with fields name, date_of_joining, and salary, and a main function that reads input for five people and prints their details. The output shows the person's name, date of joining, and salary being entered and printed.

```
2_personal.c
1 // Define a structure type struct personal that would contain person name, date of joining and
2 // salary using this structure to read this information of 5 people and print the same on screen.
3
4 #include <stdio.h>
5
6 struct personal{
7     char name[20];
8     char date_of_joining[20];
9     int salary;
10 }
11
12 int main(){
13     struct personal person[5];
14
15     for (int i = 0; i < 5; ++i){
16         printf("\nEnter person name %d : ", i+1);
17         scanf(" %[^\\n]", person[i].name);
18
19         printf("Enter date of joining : ");
20         scanf(" %[^\\n]", person[i].date_of_joining);
21
22         printf("Enter person's salary : ");
23         scanf(" %d", &person[i].salary);
24
25     }
26
27     printf("\n-----Details-----\n");
28     for (int i = 0; i < 5; ++i){
29
30         printf("\nPerson name    : %s\n", person[i].name);
31         printf("Date of joining : %s\n", person[i].date_of_joining);
32         printf("Person's salary : %d\n", person[i].salary);
33
34     }
35
36
37     return 0;
38 }
```

```
osboxes ~ > New Folder > ass_10 > struct > g 2_personal 2_personal.c
osboxes ~ > New Folder > ass_10 > struct > ./2_personal
Enter person name 1 : Naruto
Enter date of joining : 02-03-1995
Enter person's salary : 34000
Enter person name 2 : Ash
Enter date of joining : 23-11-1998
Enter person's salary : 31000
Enter person name 3 : Kirito
Enter date of joining : 30-05-2001
Enter person's salary : 33000
Enter person name 4 : May
Enter date of joining : 03-06-2002
Enter person's salary : 23000
Enter person name 5 : Asuna
Enter date of joining : 12-01-2000
Enter person's salary : 32000
-----Details-----
Person name    : Naruto
Date of joining : 02-03-1995
Person's salary : 34000
Person name    : Ash
Date of joining : 23-11-1998
Person's salary : 31000
Person name    : Kirito
Date of joining : 30-05-2001
Person's salary : 33000
Person name    : May
Date of joining : 03-06-2002
Person's salary : 23000
Person name    : Asuna
Date of joining : 12-01-2000
Person's salary : 32000
osboxes ~ > New Folder > ass_10 > struct
```

Solution 3

The screenshot shows a Linux desktop environment with several windows open:

- File Manager:** Shows a tree view of files and folders. Visible paths include `New Folder`, `ass_10`, `file handling`, `preprocessor directiv`, `storage class specifi`, `struct`, `1 std detail`, `c 1 std detail.c`, `2 personal`, `c 2 personal.c`, `3 time`, `c 3.time.c`, `4 cricket`, `c 4 cricket.c`, `5 std record`, `c 5 std record.c`, `ass_2`, `ass_3`, `ass_4`, `ass_5`, `ass_6`, `ass_7`, `ass_8`, `ass_9`, `java`, and `pow.c`. The file `3.time.c` is selected.
- Terminal:** A "Linux Lite Terminal" window titled "3.time.c" is open. It shows the C code for a `time_struct` and a `main()` function that reads user input for hours, minutes, and seconds, then prints the time in the format `16:40:51`. The terminal also displays the command line path: `osboxes ~ > New Folder > ass_10 > struct > g 3.time 3.time.c`. The output of the program is shown as:

```
osboxes ~ > New Folder > ass_10 > struct > g 3.time 3.time.c
osboxes ~ > New Folder > ass_10 > struct > ./3_time
Enter Hours : 21
Enter minute : 44
Enter second : 12
Time : 21:44:12
```

At the bottom of the terminal window, the prompt `osboxes ~ > New Folder > ass_10 > struct >` is visible.

System Tray: The system tray icons include a network connection, battery level, volume, and system status.

Solution 4

The screenshot shows a Linux desktop environment with a terminal window titled "Linux Lite Terminal -" and a code editor window titled "4_cricket.c".

Code Editor (4_cricket.c):

```
// Define a structure called cricket that will describe the following information:  
// Player name  
// Team name  
// Batting average  
// Using cricket, declare an array player with 50 elements and write a C program to read the  
// information about all the 50 players and print team-wise list containing names of players with  
// their batting average.  
  
#include <stdio.h>  
  
struct cricket{  
    char player_name[20];  
    char team_name[20];  
    float batting_avg;  
};  
  
int main(){  
    struct cricket players[50];  
    for (int i = 0; i < 3; ++i)  
    {  
        printf("\nEnter player name : ");  
        scanf(" %[^\\n]", players[i].player_name);  
        printf("Enter team name : ");  
        scanf(" %[^\\n]", players[i].team_name);  
        printf("Enter batting avg : ");  
        scanf("%f", &players[i].batting_avg);  
    }  
    printf("\n-----Details-----\n");  
    for (int i = 0; i < 3; ++i)  
    {  
        printf("\nPlayer name : %s\n", players[i].player_name);  
        printf("Team name : %s\n", players[i].team_name);  
        printf("Batting avg : %.2f\n", players[i].batting_avg);  
    }  
    return 0;  
}
```

Terminal Output:

```
osboxes: ~ > New Folder > ass_10 > struct > g 4_cricket 4_cricket.c  
osboxes: ~ > New Folder > ass_10 > struct > ./4_cricket  
  
Enter player name : Dhoni  
Enter team name : CSK  
Enter batting avg : 84.3  
  
Enter player name : Sachin  
Enter team name : INDIA  
Enter batting avg : 95.8  
  
Enter player name : Raina  
Enter team name : KK  
Enter batting avg : 65.5  
  
-----Details-----  
Player name : Dhoni  
Team name : CSK  
Batting avg : 84.30  
  
Player name : Sachin  
Team name : INDIA  
Batting avg : 95.80  
  
Player name : Raina  
Team name : KK  
Batting avg : 65.50
```

Solution 5

The screenshot shows a Linux desktop environment with several windows open:

- File Manager:** Shows a folder structure under "/Desktop/New Folder/ass_10/struct".
- Terminal:** Titled "Linux Lite Terminal", shows the command "g 5_std_record 5_std_record.c" being run, followed by the output of a C program.
- Code Editor:** Sublime Text (UNREGISTERED) showing the file "5_std_record.c". The code defines a structure "student_record" and reads data for 10 students from standard input, then prints their details.

```
4.cricket.c      5_std_record.c
1 // Design a structure student_record to contain name, branch and total marks obtained.
2 // Develop a program to read data for 10 students in a class and print them.
3
4 #include <stdio.h>
5
6 struct student_record{
7     char std_name[20];
8     char std_branch[20];
9     float std_marks;
10 }
11
12 int main(){
13     struct student_record students_data[10];
14
15     for (int i = 0; i < 3; ++i)
16     {
17         printf("\nEnter student name : ");
18         scanf("%[^\\n]", students_data[i].std_name);
19         printf("Enter branch name : ");
20         scanf("%[^\\n]", students_data[i].std_branch);
21         printf("Enter student marks : ");
22         scanf("%f", &students_data[i].std_marks);
23     }
24     printf("\n-----Details-----");
25     for (int i = 0; i < 3; ++i)
26     {
27         printf("\nStudent name : %s\n", students_data[i].std_name);
28         printf("Branch name : %s\n", students_data[i].std_branch);
29         printf("Student marks : %.2f\n", students_data[i].std_marks);
30     }
31 }
32
33 return 0;
34 }
```

Terminal Output:

```
osboxes ~ > New Folder > ass_10 > struct g 5_std_record 5_std_record.c
osboxes ~ > New Folder > ass_10 > struct ./5_std_record
Enter student name : Rahul
Enter branch name : EE
Enter student marks : 74.20
Enter student name : David
Enter branch name : CSE
Enter student marks : 87.90
Enter student name : Julie
Enter branch name : CSE
Enter student marks : 92.00
-----Details-----
Student name : Rahul
Branch name : EE
Student marks : 74.20
Student name : David
Branch name : CSE
Student marks : 87.90
Student name : Julie
Branch name : CSE
Student marks : 92.00
osboxes ~ > New Folder > ass_10 > struct
```

Preprocessor directives

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are icons for LibreOffice, a file manager, and a terminal. The terminal window is titled "Linux Lite Terminal" and contains the command "osboxes ~ New Folder > ass_10 > preprocessor directives > ./1_header_var". The output of the command is displayed below, showing the values of PI and e.

Sublime Text is open in the background, showing code for "1_header_var.c". The code includes #include <stdio.h>, #include "1_custom.h", and a main() function that prints the value of PI and e.

```
1 // Define a variable inside a header file, say "custom.h", use this header file in your
2 // program and then print the value of the variable defined in custom.h.
3
4 #include <stdio.h>
5 // inside of custom.h
6 // float PI = (float)22/7;
7 #include "1_custom.h"
8
9 int main(){
10     printf("The PI value : %f\n", PI);
11     printf("The e value : %f\n", e);
12     printf("speed of light : %d m/s\n", light_speed);
13     return 0;
14 }
15 }
```

```
osboxes ~ New Folder > ass_10 > preprocessor directives > ./1_header_var
The PI value : 3.142857
The e value : 2.718280
speed of light : 299792458 m/s
```

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are icons for LibreOffice, a file manager, and a terminal. The terminal window is titled "Linux Lite Terminal" and contains the command "osboxes ~ New Folder > ass_10 > preprocessor directives > ./2_define". The output of the command is displayed below, showing the area of a circle calculated using the formula PI * radius * radius.

Sublime Text is open in the background, showing code for "2_define.c". The code includes #include <stdio.h>, #define PI 3.14, and a main() function that prompts the user for a radius and prints the area.

```
1 // Using #define, declare a value, say PI = 3.14, and use it to calculate area of a circle,
2 // given the radius.
3
4 #include <stdio.h>
5 #define PI 3.14
6
7 int main(){
8     int radius;
9     printf("Enter the radius : ");
10    scanf("%d", &radius);
11
12    printf("\nArea of circle : %.3f\n", PI*radius*radius);
13    return 0;
14 }
```

```
osboxes ~ New Folder > ass_10 > preprocessor directives > ./2_define
Enter the radius : 4
Area of circle : 50.240
```

Solution 3

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons for system monitoring and user accounts. Below the panel, a file manager window is open, showing a directory structure for 'ass_10' containing files like '4_cricket.c', '5_std_record.c', '2_define.c', and '3_loop_macro.c'. To the right of the file manager is a terminal window titled 'Linux Lite Terminal-' with the command 'g 3_loop_macro' running. The terminal output shows several warnings from the compiler about backslash and newline separation. Below the terminal is another terminal window with the command './3_loop_macro' running, which outputs the numbers 1 through 20. At the bottom of the screen is a dock with various application icons.

```
// Write a for loop to print numbers from 1 to 20. The for loops has to be written as a
// multiline macro
#define forlo(x, y)
{
    for(int i=x; i<=y; ++i) \
        printf("%d\n", i);
    printf("\n");
}

int main(){
    forlo(1, 20);
    return 0;
}
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
osboxes ... > New Folder > ass_10 > preprocessor directives > g 3_loop_macro 3_
loop_macro.c
3_loop_macro.c:5:24: warning: backslash and newline separated by space
      #define forlo(x, y) \
                                         ^
3_loop_macro.c:6:10: warning: backslash and newline separated by space
  6 |
      {
3_loop_macro.c:8:25: warning: backslash and newline separated by space
  8 |     printf("%d\n", i); \
                                         ^
osboxes ... > New Folder > ass_10 > preprocessor directives > ./3_loop_macro
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Solution 4

The screenshot shows a Linux desktop environment with a dark theme, similar to the previous one. In the top panel, there are icons for system monitoring and user accounts. Below the panel, a file manager window is open, showing a directory structure for 'ass_10' containing files like '4_cricket.c', '5_std_record.c', '2_define.c', '3_loop_macro.c', and '4_ifdef.c'. To the right of the file manager is a terminal window titled 'Linux Lite Terminal-' with the command 'g 4_ifdef' running. The terminal output shows several warnings from the compiler about undefined symbols. Below the terminal is another terminal window with the command './4_ifdef' running, which outputs the message 'var is undefined'. At the bottom of the screen is a dock with various application icons.

```
// Give example usage of conditional compilation using #ifdef, #ifndef and #undef.
#include <stdio.h>
#define VAR
#define VAR1

int main()
{
    //return true if this macro is defined
    #ifdef VAR
    printf("var is defined\n");
    #else
    printf("var is not defined\n");
    #endif

    //undefines a preprocessor macro
    #undef VAR

    #ifdef VAR
    printf("var is defined\n");
    #else
    printf("var is undefined\n");
    #endif

    //return true if this macro is not defined
    #ifndef VAR1
    printf("var1 is not defined\n");
    #else
    printf("var1 is defined\n");
    #endif
}

return 0;
}
```

```
.c
osboxes ... > New Folder > ass_10 > preprocessor directives > g 4_ifdef 4_ifdef
osboxes ... > New Folder > ass_10 > preprocessor directives > ./4_ifdef
var is undefined
var is undefined
var1 is undefined
osboxes ... > New Folder > ass_10 > preprocessor directives >
```

Solution 5

The screenshot shows a Linux desktop environment with several windows open:

- File Manager:** Shows a folder structure under "New Folder" containing files like 4_cricket.c, 5_std_record.c, 2_define.c, 3_loop_macro.c, and 5_concat.c.
- Terminal:** A "Linux Lite Terminal" window titled "Linux Lite Terminal -". It displays the following command and output:

```
osboxes ... > New Folder > ass_10 > preprocessor directives g 5_concat 5_concat.c
osboxes ... > New Folder > ass_10 > preprocessor directives ./5_concat
123456
osboxes ... > New Folder > ass_10 > preprocessor directives
```
- Code Editor:** A "Sublime Text (UNREGISTERED)" window titled "5_concat.c" showing C code. The code defines a macro CAT that concatenates two numbers into a single string using the ## operator. It includes a main function that prints the result of CAT(123, 456).

The desktop bar at the bottom shows icons for PDF, assignments, and other system applications.

```
4_cricket.c      5_std_record.c    2_define.c      3_loop_macro.c    5_concat.c
1 // Concat two numbers to create a single number using ## operator:
2
3 #include <stdio.h>
4 #define CAT(x, y)printf("\n%d\n", x##y);
5
6 int main(){
7     CAT(123, 456)
8
9     return 0;
10}
11}
```

Storage Class Specifier

Solution 1

The screenshot shows a Linux desktop environment with several windows open:

- A file manager window showing a directory structure with files like 1_static.c, 2_extern.c, 3_localvar.c, etc.
- A terminal window titled "Linux Lite Terminal" showing the command: `g 1_static 1_static`.
- A code editor window titled "1_static.c" containing C code that uses the `static` keyword to declare a global variable `count` and a function `fun` that increments it each time it is called. The code also includes a `main` function that calls `fun` three times and prints its value.
- A status bar at the bottom indicates "Line 20, Column 15".
- A dock at the bottom contains icons for various applications like LibreOffice, File Manager, Terminal, and others.

Solution 2

The screenshot shows a Linux desktop environment with several windows open:

- A file manager window showing a directory structure with files like 1_static.c, 2_extern.c, 3_localvar.c, etc.
- A terminal window titled "Linux Lite Terminal" showing the command: `g bin 2_extern.c 2_extern2.c`. It also displays a warning message: `2_extern2.c:3:12: warning: 'var' initialized and declared 'extern'`.
- A code editor window titled "2_extern.c" containing C code that declares an `extern int var;` in the first file and defines it with a value of 420 in the second file. The code also includes a `main` function that prints the value of `var`.
- A status bar at the bottom indicates "10 characters selected".
- A dock at the bottom contains icons for various applications like LibreOffice, File Manager, Terminal, and others.

Solution 3

The screenshot shows a Linux desktop environment with several windows open:

- File Manager:** Shows a folder structure under "New Folder". One file, "3_localvar.c", is currently selected.
- Terminal:** Two tabs are visible:
 - The first tab shows the command: `osboxes ~ > New Folder > ass_10 > storage class specifier > g 3_localvar 3_localvar.c`. The output shows:

```
fun1 -> var = 4
fun2 -> var = 9
```
 - The second tab shows the command: `osboxes ~ > New Folder > ass_10 > storage class specifier`.
- Sublime Text:** A code editor window titled "3_localvar.c (New Folder) - Sublime Text (UNREGISTERED)". It displays the following C code:

```
1 static.c      2 extern.c      3 localvar.c      2 extern2.c
 1 // Can two functions have variables with the same name?
 2 // Give example of they both have different scopes.
 3
 4 // Yes, two or more function can have variable with the same name,
 5 // because that same name variable always to local to it specific function.
 6 // which means that var inside func does exist or scope outside the func.
 7 #include <stdio.h>
 8
 9 int fun1(){
10     // variable var local to fun1
11     int var = 4;
12     printf("fun1 -> var = %d\n", var);
13     return 0;
14 }
15
16 int fun2(){
17     // variable var local to fun2
18     int var = 9;
19     printf("fun2 -> var = %d\n", var);
20     return 0;
21 }
22
23 int main(){
24     fun1();
25     fun2();
26     return 0;
27 }
```

File handling

Solution 1

The screenshot shows a Linux desktop environment with several windows open. In the foreground, there is a terminal window titled 'Linux Lite Terminal' with the command 'cd ..;/file\ handling/g_1_write 1_write.c' and the output 'Input the number of lines to be written : 3'. Below it, another terminal window shows the contents of 'output.txt' with three lines of text: 'Line 1', 'Line 2', and 'Line 3'. In the background, there is a code editor window titled 'Sublime Text (UNREGISTERED)' showing a C program named '1_write.c'. The code reads user input for the number of lines to write and then writes those lines to a file named 'output.txt'. The code editor has syntax highlighting for C and shows line numbers from 1 to 38.

```
1 // Write a program in C to write multiple lines in a text file.
2 // Test Data :
3 // Input the number of lines to be written : 4
4 // The lines are
5 // test line 1
6 // test line 2
7 // test line 3
8 // Expected Output :
9 // The content of the file test.txt is :
10 // test line 1
11 // test line 2
12 // test line 3
13
14 #include <stdio.h>
15 #include <stdlib.h>
16
17 int main(){
18     FILE *file = fopen("output.txt", "w");
19
20     if (file == NULL){
21         printf("fail to open!");
22         return -1;
23     }
24     char s[100]; int line;
25
26     printf("Input the number of lines to be written : ");
27     scanf("%d", &line);
28
29     for(int i=0; i<line-1; i++){
30         fgets(s, sizeof(s), stdin);
31         fputs(s, file);
32     }
33
34     fclose(file);
35     return 0;
36 }
37
38
```

Soultion 2

The screenshot shows a Linux desktop environment with several windows open. In the foreground, there is a terminal window titled 'Linux Lite Terminal' with the command 'cd ..;/file\ handling/g_2_read 2_read.c' and the output 'Input the file name to be opened : test.txt'. Below it, another terminal window shows the contents of 'output.txt' with three lines of text: 'Line 1', 'Line 2', and 'Line 3'. In the background, there is a code editor window titled 'Sublime Text (UNREGISTERED)' showing a C program named '2_read.c'. The code reads the contents of a file named 'test.txt' and stores the lines into an array. The code editor has syntax highlighting for C and shows line numbers from 1 to 33.

```
1 // Write a program in C to read the file and store the lines into an array.
2 // Test Data :
3 // Input the file name to be opened : test.txt
4 // Expected Output :
5 // The content of the file output.txt are :
6 // Line 1
7 // Line 2
8 // Line 3
9
10 #include <stdio.h>
11
12 int main(){
13     FILE *file;
14     char s[50], c;
15
16     file = fopen("output.txt", "r");
17
18     if(file == NULL){
19         printf("File not found\n");
20         return -1;
21     }
22     int i=0;
23     while ((c = getc(file)) != EOF){
24         s[i] = c; i++;
25     }
26
27     printf("%s\n", s);
28
29     return 0;
30 }
31
32
33
```

Soultion 3

The screenshot shows a Linux desktop environment with several windows open:

- File Manager:** Shows a folder structure under "/Desktop/Desktop/New Folder/ass_10/file handling".
- Sublime Text (UNREGISTERED):** An open file named "3_count_words.c" containing C code to count words and characters in a file.
- Linux Lite Terminal:** A terminal window with the following session:

```
osboxes ... > New Folder > ass_10 > file handling > gcc -o 3_count_words 3_count_words.c
osboxes ... > New Folder > ass_10 > file handling > ./3_count_words
test line 1
test line 2
test line 3
test line 4

The number of words in the file test.txt are : 12
The number of characters in the file test.txt are : 36
```

Soultion 4

The screenshot shows a Linux desktop environment with several windows open:

- File Manager:** Shows a folder structure under "/Desktop/Desktop/New Folder/ass_10/file handling".
- Sublime Text (UNREGISTERED):** An open file named "4_copy_file.c" containing C code to copy a file from source to destination.
- Linux Lite Terminal:** A terminal window with the following session:

```
osboxes ... > New Folder > ass_10 > file handling > gcc -o 4_copy_file 4_copy_file.c
osboxes ... > New Folder > ass_10 > file handling > ./4_copy_file
Enter source file name : 4.test.txt
Enter destination file name : 4_test2.txt

test line 1
test line 2
test line 3
test line 4
```

Soultion 5

The screenshot shows a Linux desktop environment with several windows open:

- File Manager:** Shows a tree view of files and folders, including 'New Folder', 'ass_10' (containing 'file handling', '1 write', '2 read', etc.), '3_count_words.c', '4_copy_file.c', '5_replace_data.c', and 'test.txt'.
- Code Editor:** Displays the source code for '5_replace_data.c'. The code handles file operations like reading from 'source.txt', writing to 'temp.txt', and replacing specific lines. It includes error checking for file existence and EOF.
- Terminal:** Shows a terminal window titled 'Linux Lite Terminal'. The user runs 'gcc -o 5_replace_data 5_replace_data.c' to compile the program. Then, they input the file name ('test.txt'), content ('Hello dear'), and line number (2) to replace. The terminal output shows the original file content, the replacement command, and the modified file content ('Hello line 2').

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

int main()
{
    FILE *file1, *file2;
    int lno, linectr = 0;
    char s[50], source[20];
    char newln[50], temp[] = "temp.txt";

    printf(" Input the file name to be opened : ");
    fgets(source, 20, stdin);
    source[strlen(source) - 1] = '\0';
    file1 = fopen(source, "r");
    if (!file1) {
        printf(" File not found\n");
        return -1;
    }

    file2 = fopen(temp, "w");
    if (!file2) {
        printf(" File not found\n");
        fclose(file1);
        return -1;
    }

    printf(" Input the content for the File : ");
    fgets(newln, 50, stdin);

    printf(" Input the line no you want to replace : ");
    scanf("%d", &lno);
    lno++;

    while ((!feof(file1)))
    {
        strcpy(s, "\0");
        fgets(s, 50, file1);
        if (!feof(file1))
        {
            linectr++;
            if (linectr != lno)
            {
                fprintf(file2, "%s", s);
            }
            else
            {
                fprintf(file2, "%s", newln);
            }
        }
    }
    fclose(file1);
    fclose(file2);
}
```

Line 16, Column 1 Tab Size: 4