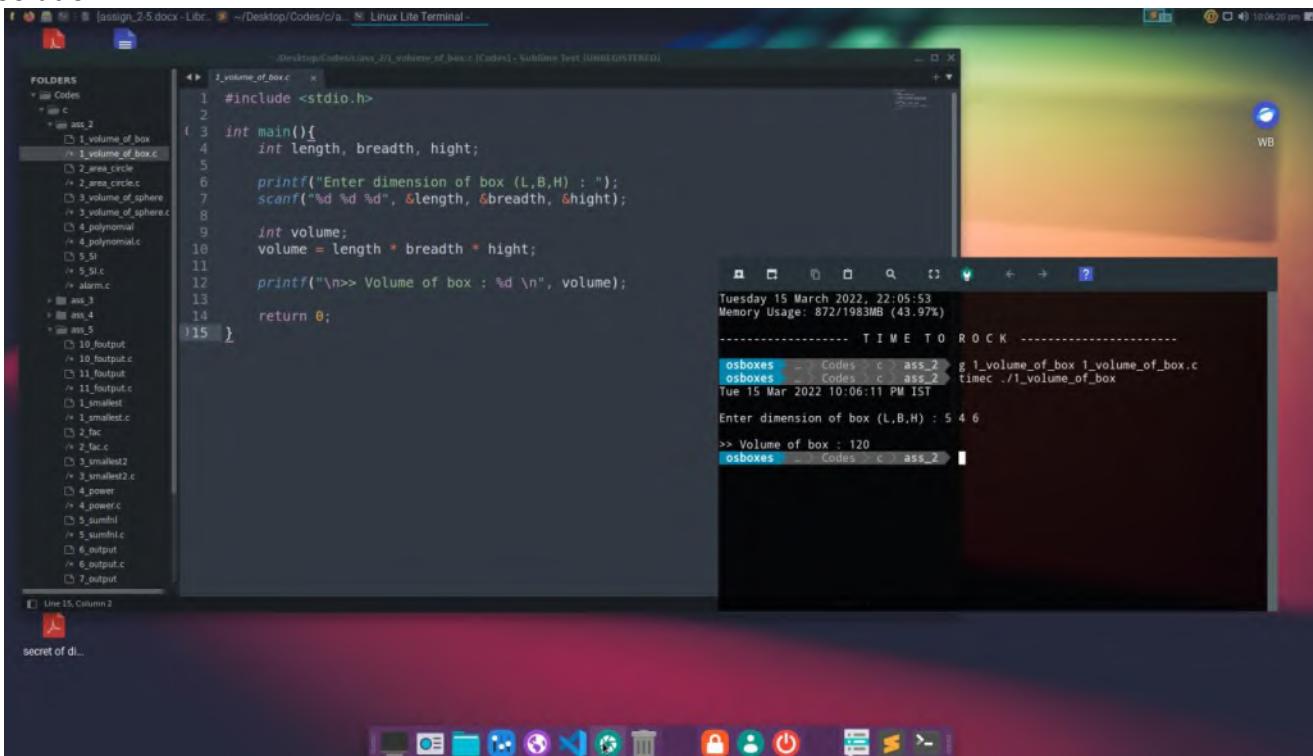


Roll no: 220350320070
Name: MD SAIF
PG-DAC March 2022
Module 01: Basic Programming Concepts

DAY 2

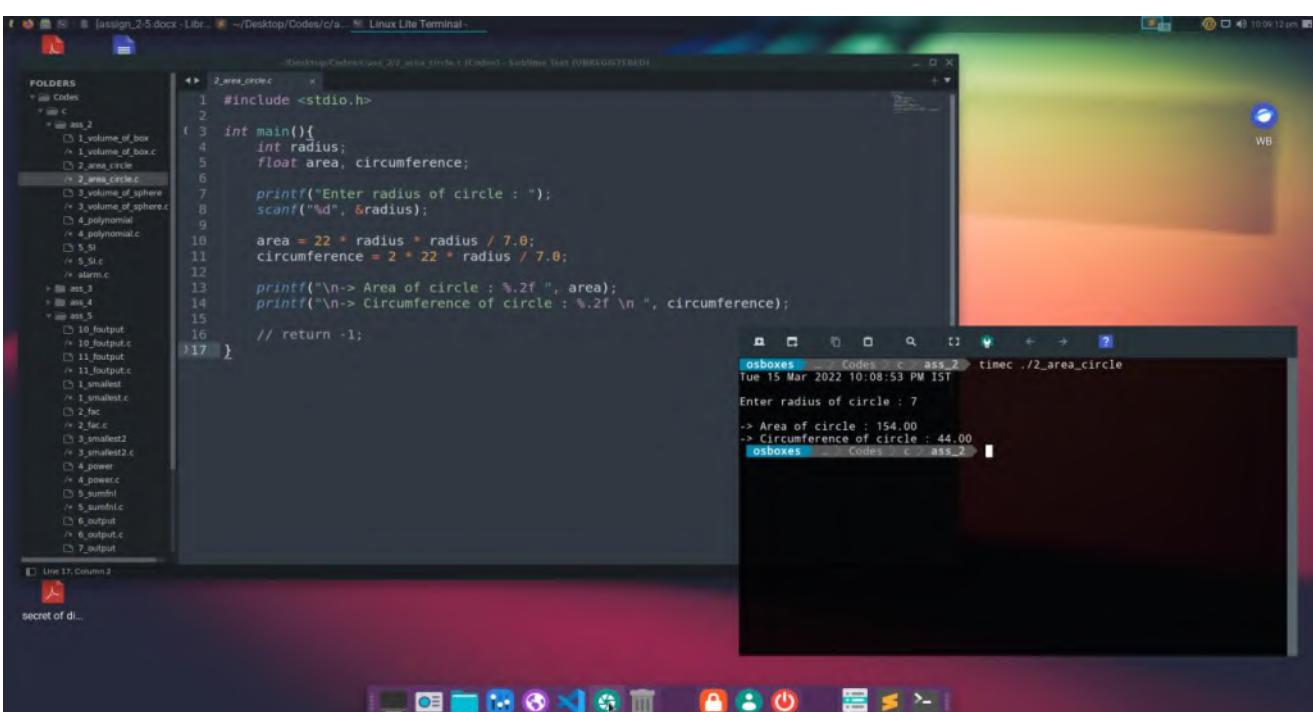
Solution 1:



A screenshot of a Linux desktop environment. On the left, there's a file manager window showing a directory structure under 'Codes'. In the center, a terminal window titled 'Linux Lite Terminal' is open, displaying the code for '1_volume_of_box.c'. The code calculates the volume of a box given its length, breadth, and height. To the right of the terminal is a Sublime Text editor window showing the same code. The desktop background is a colorful abstract image.

```
#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



A screenshot of a Linux desktop environment, similar to the one above. A terminal window titled 'Linux Lite Terminal' shows the code for '2_area_circle.c'. This program calculates the area and circumference of a circle given its radius. The code uses the formula $\text{area} = 22 * \text{radius} * \text{radius} / 7.0$ and $\text{circumference} = 2 * 22 * \text{radius} / 7.0$. The Sublime Text editor window on the right also contains this code. The desktop background is the same colorful abstract image.

```
#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 terminal window and a code editor window.

Code Editor: The code editor window displays a C program named `3_volume_of_sphere.c`. The code calculates the volume of a sphere given its radius. It includes a `#define PI` for pi and uses `scanf` and `printf` functions.

```
#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;
}
```

Terminal: The terminal window shows the output of the program. It prompts for the radius, receives input of 7, and then prints the volume as 1437.33.

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

Solution 4

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor: The code editor window displays a C program named `4_polynomial.c`. The program calculates the value of a polynomial $y = 6x^5 + 5x^4 - 4x^3 + 3x^2 + 2x + 1$ for a given x .

```
#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;
}
```

Terminal: The terminal window shows the output of the program. It prompts for the value of x , receives input of 1, and then prints the value of y as 1437.33. It also shows some compilation errors related to undefined references to `'pow'`.

```
osboxes ~/Codes/c/ass_2 g 4_polynomial 4_polynomial
/usr/bin/ld: /usr/lib/gcc/x86_64-pc-linux-gnu/10.2.0/../../../../lib/crt0.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+0x93): 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 l g 4_polynomial 4_polynomial
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 = 1437.33
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
```

Solution 5

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Code Editor: Sublime Text (UNREGISTERED) showing file `2_area_code.c`:

```
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 }
19 }
```

Terminal: osboxes ~ Codes c ass_2 timetc ./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 timetc ./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 timetc ./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 'osboxes' and shows the command 'timec ./1_operators'. It displays the output of the program, which asks for two numbers (4 and 2) and then prints various arithmetic operations: addition, subtraction, multiplication, division, and modulus.

```
#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 = %d\n", a, b, a%b);
    return 0;
}
```

```
osboxes ~ Codes c ass_2 cd ../ass_3/
osboxes ~ Codes c ass_3 timec ./1_operators
Tue 15 Mar 2022 10:21:51 PM IST
Enter two numbers : 4 2
4 + 2 = 6.00
4 - 2 = 2.00
4 * 2 = 8.00
4 / 2 = 2.00
4 % 2 = 0
osboxes ~ Codes c ass_3 timec ./1_operators
Tue 15 Mar 2022 10:21:06 PM IST
Enter two numbers : 7 3
7 + 3 = 10.00
7 - 3 = 4.00
7 * 3 = 21.00
7 / 3 = 2.33
7 % 3 = 1
osboxes ~ Codes c ass_3
```

solution 2

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window is titled 'osboxes' and shows the command 'timec ./2_bitwise_operators'. It displays the output of the program, which asks for two numbers (6 and 2) and then prints various bitwise operations: NOT, AND, OR, XOR, left shift, right shift, and modulus.

```
#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;
}
```

```
osboxes ~ Codes c ass_3 g 2_bitwise_operators 2_bitwise_operators.c
osboxes ~ 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 ~ Codes c ass_3
```

Solution 3

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window is titled 'Linux Lite Terminal' and contains the following text:

```
osboxes: ~/Codes/c/ ass_3 g 2_bitwise_operators 2_bitwise_operators.c
osboxes: ~/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: ~/Codes/c/ ass_3 g 3_add_ass 3_add_ass.c
osboxes: ~/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: ~/Codes/c/ ass_3
```

The code editor window shows a file named '3_add_ass.c' with the following content:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a, b;
6
7     printf("Enter two numbers : ");
8     scanf("%d%d", &a, &b);
9
10    a += b;
11
12    printf("\nAddition: a + b = %d\n", a);
13    return 0;
14 }
```

Solution 4

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window is titled 'Linux Lite Terminal' and contains the following text:

```
osboxes: ~/Codes/c/ ass_3 g 2_bitwise_operators 2_bitwise_operators.c
osboxes: ~/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: ~/Codes/c/ ass_3 g 3_add_ass 3_add_ass.c
osboxes: ~/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: ~/Codes/c/ ass_3 g 4_maxOf2 4_maxOf2.c
osboxes: ~/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: ~/Codes/c/ ass_3
```

The code editor window shows a file named '4_maxOf2.c' with the following content:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a, b;
6
7     printf("Enter two numbers : ");
8     scanf("%d%d", &a, &b);
9
10    if(a>b) printf("\nmaximum is %d\n", a); else printf("\nmaximum is %d\n", b);
11    return 0;
12 }
```

Day 4

Solution 1:

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window on the right displays the output of a C program named `l_positice_nagetive.c`. The code in the editor handles floating-point numbers based on their sign. It includes logic to check if a number is positive, negative, or zero, and prints the corresponding message.

```
#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 negative\n", val);
    else
        printf("Number is zero\n");

    return 0;
}
```

Terminal Output:

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

Solution 2

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window on the right displays the output of a C program named `2_greater_num.c`. The code in the editor finds the maximum of three input numbers. It uses nested conditional statements to compare three variables (`val1`, `val2`, `val3`) and determine which one is the greatest.

```
#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 : val2>val3 ? val2 : val3;

    printf("\n%.0f is greatest in all\n", max);

    return 0;
}
```

Terminal Output:

```
osboxes:~/Codes/c/ass_4> g 2_greater_num.c
osboxes:~/Codes/c/ass_4> timec ./2_greater_num
Wed 16 Mar 2022 12:30:00 AM IST
Enter value : 8
8.00 is positive
osboxes:~/Codes/c/ass_4> timec ./2_greater_num
Wed 16 Mar 2022 12:30:18 AM IST
Enter value : 0
Number is zero
osboxes:~/Codes/c/ass_4> timec ./2_greater_num
Wed 16 Mar 2022 12:30:25 AM IST
Enter value : -4
-4.00 is negative
osboxes:~/Codes/c/ass_4> g 2_greater_num 2_greater_num.c
osboxes:~/Codes/c/ass_4> 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> 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>
```

solution 3

A screenshot of a Linux desktop environment. On the left, there is a file browser window titled 'Codes' showing various C source files. On the right, there is a terminal window titled 'Linux Lite Terminal'. The terminal shows the command 'timec ./3_smallest_num' being run, followed by the output: 'Enter 3 numbers : 6 3 9' and '3 is smallest number'. Below this, another run of the command is shown with the output: 'Enter 3 numbers : 4 8 0' and '0 is smallest number'.

```
#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

A screenshot of a Linux desktop environment. On the left, there is a file browser window titled 'Codes' showing various C source files. On the right, there is a terminal window titled 'Linux Lite Terminal'. The terminal shows the command 'timec ./3_smallest_num' being run, followed by the output: 'Enter 3 numbers : 6 3 9' and '3 is smallest number'. Below this, another run of the command is shown with the output: 'Enter 3 numbers : 4 8 0' and '0 is smallest number'. At the bottom of the terminal, it says 'CDAC 5'.

```
#include <stdio.h>
int main()
{
    int i;
    if( a, b ,c) // in this case condition value taken from c i.e value of c take first
    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(i==0,1)           // 5 will assing 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. On the left, there is a file browser window titled 'Calculator' showing a directory structure under '/Desktop/Codes/c/a'. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'g 5_find_output 5_find_output.c' and the output '38'. On the right, there is another terminal window titled 'Linux Lite Terminal' with the command 'timec ./5_find_output' and the output '38'. The desktop has a dark theme with icons for various applications like calculator, file manager, and terminal.

```
#include <stdio.h>
int main()
{
    int a=12, b=18, c=7, d=4, result;
    result = a + a * -b/c % d + c * d;
    // r = 12 + 12 * -18 / 7 % 4 + 7 * 4
    // r = 12 - 216 / 7 % 4 + 28
    // r = 12 - 30 % 4 + 28
    // r = 12 - 2 + 28
    // r = 10 + 28
    // r = 38
    printf("%d\n", result);
    return 0;
}
```

solution 6

A screenshot of a Linux desktop environment. On the left, there is a file browser window titled 'Calculator' showing a directory structure under '/Desktop/Codes/c/a'. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'g 6_find_output 6_find_output.c' and the output '11'. On the right, there is another terminal window titled 'Linux Lite Terminal' with the command 'timec ./6_find_output' and the output '11'. The desktop has a dark theme with icons for various applications like calculator, file manager, and terminal.

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

solution 7

A screenshot of a Linux desktop environment. On the left, there is a file browser window titled 'Calculator' showing a directory structure under '/Desktop/Codes/c/a'. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'timec ./7_find_output' and its output: '5'. To the right of the terminal is a code editor window titled 'Sublime Text (UNREGISTERED)' showing a C program named '7_find_output.c'. The code contains logic to calculate the value of b based on the value of h.

```
#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;
}
```

solution 8

A screenshot of a Linux desktop environment. On the left, there is a file browser window titled 'Calculator' showing a directory structure under '/Desktop/Codes/c/a'. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'timec ./8_find_output' and its output: 'false'. To the right of the terminal is a code editor window titled 'Sublime Text (UNREGISTERED)' showing a C program named '8_find_output.c'. The code contains an if-else statement that prints 'true' if x is 0 and 'false' if x is 1.

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

solution 9

A screenshot of a Linux desktop environment. On the left, there's a file browser window titled 'Codes' showing various C files. In the center, there's a terminal window titled 'Linux Lite Terminal' with the command 'timec ./9_find_output' running. The output shows:

```
osboxes ~ Codes c ass_4 g 9_find_output 9_find_output.c
Wed 16 Mar 2022 12:49:07 AM IST
false
osboxes ~ Codes c ass_4
```

The terminal window has a dark background with light-colored text. The desktop icons at the bottom include a calculator, file manager, terminal, and system tray icons.

Solution 10

A screenshot of a Linux desktop environment. On the left, there's a file browser window titled 'Codes' showing various C files. In the center, there's a terminal window titled 'Linux Lite Terminal' with the command 'timec ./10_find_output' running. The output shows:

```
osboxes ~ Codes c ass_4 g 10_find_output 10_find_output.c
Wed 16 Mar 2022 12:52:55 AM IST
Delhi
a=2 b=0 c=-1 d=0
osboxes ~ Codes c ass_4
```

The terminal window has a dark background with light-colored text. The desktop icons at the bottom include a calculator, file manager, terminal, and system tray icons.

Day5

Solution 1:

A screenshot of a Linux desktop environment. On the left, a terminal window titled 'Linux Lite Terminal' shows the command 'timec ./1_smallest' and its output: 'Enter the three values : 8 4 7' and '4 is smallest'. On the right, a code editor window titled 'Calculator' displays a C program named '1_smallest.c'. The program prompts the user for three integers, compares them using an if-else ladder, and prints the smallest value. The code is well-formatted with proper indentation and comments.

```
#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;
}
```

Solution 2

A screenshot of a Linux desktop environment. On the left, a terminal window titled 'Linux Lite Terminal' shows the command 'timec ./2_fac' and its output: 'Enter a number : 7' and 'Factorial of 7 : 5040'. On the right, a code editor window titled 'Calculator' displays a C program named '2_fac.c'. The program calculates the factorial of a given number by iteratively multiplying it with decreasing values until it reaches 1. The code includes a check for negative numbers.

```
#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;
}
```

Solution 3

A screenshot of a Linux desktop environment. On the left, a file manager window shows a directory structure under 'Codes' containing various C files like 1_smallest.c, 2_fac.c, etc. In the center, a terminal window titled 'Linux Lite Terminal' is open, displaying the output of a program. The terminal shows the following interaction:

```
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.
```

Solution 4

A screenshot of a Linux desktop environment. On the left, a file manager window shows a directory structure under 'Codes' containing various C files like 1_smallest.c, 2_fac.c, etc. In the center, a terminal window titled 'Linux Lite Terminal' is open, displaying the output of a program. The terminal shows the following interaction:

```
osboxes ~/Codes/c/ass_5> g 4_power 4_power.c
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
Exponent should greater than equal to 0
```

Solution 5

A screenshot of a Linux desktop environment. On the left, a file manager window shows a directory structure under 'Codes'. In the center, a terminal window titled 'Linux Lite Terminal' displays the following C code:

```
#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: \n", temp);
    printf("%d + %d = %d \n", first_num, last_num, first_num+last_num);
    return 0;
}
```

To the right of the terminal, another terminal window shows the output of the program:

```
osboxes ~ Codes c ass_5 g 5_sumfnl
osboxes ~ Codes c ass_5 timec ./5_sumfnl
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
```

Solution 6

A screenshot of a Linux desktop environment. On the left, a file manager window shows a directory structure under 'Codes'. In the center, a terminal window titled 'Linux Lite Terminal' displays the following C code:

```
#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;
}
```

To the right of the terminal, another terminal window shows the output of the program:

```
osboxes ~ Codes c ass_5 g 6_output
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
```

Solution 7

The screenshot shows a Linux desktop environment with a dark theme. On the left, a file explorer window titled 'Codes' is open, showing a directory structure with various C files. In the center, there are two terminal windows. The top terminal window has tabs for 'assignment_2-5.pdf', 'Calculator', and 'Linux Lite Terminal'. It displays the code for problem 7, which is a C program that prints 'True' if the first argument is 1 and 'False' otherwise. The bottom terminal window also has tabs for 'assignment_2-5.pdf', 'Calculator', and 'Linux Lite Terminal'. It shows the command 'timec ./7_output' being run, with the output 'True' and 'False' displayed. The desktop bar at the bottom includes icons for file operations, system status, and network.

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

```
osboxes ~ > Codes > ass_5 > g 7_output 7_output.c
osboxes ~ > Codes > ass_5 > timec ./7_output
Wed 16 Mar 2022 01:38:32 AM IST
True
False
osboxes ~ > Codes > ass_5 >
```

Solution 8

The screenshot shows a Linux desktop environment with a dark theme. On the left, a file explorer window titled 'Codes' is open, showing a directory structure with various C files. In the center, there are two terminal windows. The top terminal window has tabs for 'assignment_2-5.pdf', 'Calculator', and 'Linux Lite Terminal'. It displays the code for problem 8, which is a C program that calculates the value of πr^2 given p and t . The bottom terminal window also has tabs for 'assignment_2-5.pdf', 'Calculator', and 'Linux Lite Terminal'. It shows the command 'timec ./8_output' being run, with the output '1500.00' displayed. The desktop bar at the bottom includes icons for file operations, system status, and network.

```
#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;
}
```

```
osboxes ~ > Codes > ass_5 > g 7_output 7_output.c
osboxes ~ > Codes > ass_5 > timec ./7_output
Wed 16 Mar 2022 01:38:32 AM IST
True
False
osboxes ~ > Codes > ass_5 > g 8_output 8_output.c
osboxes ~ > Codes > ass_5 > timec ./8_output
Wed 16 Mar 2022 01:39:54 AM IST
1500.00
osboxes ~ > Codes > ass_5 >
```

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 network connection. Below it, the date and time are displayed as "Wed 16 Mar 2022 01:45:02 AM IST".

On the left side, there is a file browser window titled "Linux Lite Terminal" showing a directory structure under "/Desktop/Codes/c/ass_5". The current file being edited is "9_output.c". The code in the editor is:

```
#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;
}
```

To the right of the file browser is a terminal window titled "Linux Lite Terminal" with the command "timec ./9_output" entered. The output of the command is "a=65".

The bottom of the screen features a dock with various icons, including a file manager, terminal, and system settings.

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 network connection. Below it, the date and time are displayed as "Wed 16 Mar 2022 01:46:11 AM IST".

On the left side, there is a file browser window titled "Linux Lite Terminal" showing a directory structure under "/Desktop/Codes/c/ass_5". The current file being edited is "10_output.c". The code in the editor is:

```
#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;
}
```

To the right of the file browser is a terminal window titled "Linux Lite Terminal" with the command "timec ./10_output" entered. The output of the command is "5 7".

The bottom of the screen features a dock with various icons, including a file manager, terminal, and system settings.

Solution 11

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for the date and time (Wed 16 Mar 2022 01:49:55 AM IST). Below the tray, there are two terminal windows. The left terminal window is titled "Linux Lite Terminal -" and contains the code for "11_foutput.c". The right terminal window is also titled "Linux Lite Terminal -" and shows the command "g 11_foutput 11_foutput.c" being run, followed by the output "0" and "1". A file manager window is open in the background, showing a directory structure under "/Desktop/Codes/c/ass_5". The file "11_foutput.c" is selected. The desktop bar at the bottom has various icons for system functions like power, network, and file operations.

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int i=0,j=1;
6     printf("%d\n", i++*j++);
7     printf("%d %d\n", i, j); // i=1 j=1
8
9     return 0;
10 }
```

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

Solution 12

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for the date and time (12:13:44 am). Below the tray, there are two terminal windows. The left terminal window is titled "Linux Lite Terminal -" and contains the code for "12_count_digit.c". The right terminal window is also titled "Linux Lite Terminal -" and shows the command "g 12_count_digit 12_count_digit.c" being run, followed by the prompt "Enter the number :". The user enters "345", and the output shows "the number 345 contains 3 digits.". The desktop bar at the bottom has various icons for system functions like power, network, and file operations.

```
1 // Write a C program to count total number of digits of an
2 // Integer number (n).
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 count=0, tempnum;
14    do
15    {
16        count++;
17        } while (num!=10);
18
19    printf("the number %d contains %d digits.\n", temp, count);
20
21    return 0;
22 }
```

```
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 terminal window and a code editor window.

Code Editor (Sublime Text):

```
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 }
```

Terminal (Linux Lite Terminal):

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

Solution 14

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Sublime Text):

```
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
18    printf("Sum of the series is : %.2f\n", sum);
19    return 0;
20 }
```

Terminal (Linux Lite Terminal):

```
osboxes ~ Desktop / New Folder ass_5 g 14_sumofseries 14_sumofseries.c
osboxes ~ Desktop / New Folder ass_5 ./14_sumofseries
Enter the number : 5
Sum of the series is : 2.28
osboxes ~ Desktop / New Folder ass_5 ./14_sumofseries
Enter the number : 8
Sum of the series is : 2.72
osboxes ~ Desktop / New Folder ass_5 ./14_sumofseries
Enter the number : 1
Sum of the series is : 1.00
osboxes ~ Desktop / New Folder ass_5 ./14_sumofseries
Enter the number : 2
Sum of the series is : 1.50
osboxes ~ Desktop / New Folder ass_5
```

Solution 15

Solution 16

The screenshot shows a Linux desktop environment with a green and blue gradient background. In the top right corner, there are icons for file management and system status. The bottom of the screen features a dock with various application icons.

Code Editor:

- File menu: Open, Save, Save As, Find, Replace, Cut, Copy, Paste, Select All, Find in Files, Go To, Preferences, Exit.
- Code pane:

```
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 }
```
- Bottom status bar: Line 1, Column 1

Terminal:

- Title bar: Linux Lite Terminal.
- Content:

```
osboxes ~ Desktop New Folder ass_5 g 16_perfect_number 16_perfect_number.c
osboxes ~ Desktop New Folder ass_5 ./16_perfect_number
Enter the number : 7
7 is not a perfect number
osboxes ~ Desktop New Folder ass_5 ./16_perfect_number
Enter the number : 6
6 is a perfect number
osboxes ~ Desktop New Folder ass_5
```

Solution 17

The screenshot shows a Linux desktop environment with several windows open. In the top right corner, there is a system tray icon for 'WB' and a 'New Folder' icon. The bottom of the screen features a dock with various application icons.

The main focus is on two windows:

- Sublime Text (UNREGISTERED)**: A code editor window titled "17_gcd.c" containing C code for finding the Greatest Common Divisor (GCD) of two numbers. The code uses a for loop to iterate through possible divisors from 1 to the size of the input numbers. It prints the GCD and returns 0 if no divisor is found.
- Linux Lite Terminal**: A terminal window showing the execution of the program. It prompts the user for two numbers (6 and 12), prints the GCD (6), and then prompts for another pair (9 and 81), printing the GCD (9). Finally, it prompts for a third pair (4 and 70), printing the GCD (4).

```
17_gcd.c
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 4
osboxes ~ Desktop New Folder ass_5
```

DAY 6

Solution 1

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a folder structure with files like 'ass_10', 'ass_2', 'ass_3', etc. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'osboxes ~ % ./1_armstrong' and its output: '371 is an armstrong number.'. To the right of the terminal is a code editor window titled '1_armstrong.c' containing C code for finding Armstrong numbers. The code includes comments explaining the logic of counting digits and calculating the sum of cubes of digits.

```
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     }
19 //calculate armstrong number from input number
20     num = temp;
21     while(num){
22         arm += pow(num%10, p);
23         num/=10;
24     }
25
26     //print to tell number is armstrong or not
27     if(temp == arm)
28         printf("%d is an armstrong number.\n", temp);
29     else
30         printf("%d is not an armstrong number.\n", temp);
31
32     return 0;
33 }
```

Solution 2

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a folder structure with files like 'ass_10', 'ass_2', 'ass_3', etc. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'osboxes ~ % ./2_palindrome' and its output: '121 is palindrome'. To the right of the terminal is a code editor window titled '2_palindrome.c' containing C code for checking if a number is a palindrome. The code uses a reverse-and-compare approach where it reverses the number and compares it with the original.

```
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!=0){
17        p*=10;
18
19        num = temp;
20
21        //reversing the number
22        while(num){
23            rem = num%10;
24            pal += rem*p;
25            p/=10;
26            num/=10;
27        }
28
29        //comparing reverse number with original
30        if(pal == temp)
31            printf("%d is palindrome\n", pal);
32        else
33            printf("%d is not a palindrome\n", pal);
34    }
35 }
```

Solution 3

A screenshot of a Linux desktop environment. 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 corner, there is a dock with icons for 'pdf' and 'assignments'. The terminal window shows the output of running the program '3_prime.c' to check if 101 is prime. The code editor window shows the C program '3_prime.c' which prompts the user to enter a number and checks if it is prime.

```
3_prime.c
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 is not a prime\n", num);
21
22    }
23 }
```

```
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 is not a prime
osboxes ~ Desktop New Folder ass_6 ./3_prime
Enter the number : 32
32 is 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

A screenshot of a Linux desktop environment. 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 corner, there is a dock with icons for 'pdf' and 'assignments'. The terminal window shows the output of running the program '4_prime10.c' to find all prime numbers from 1 to 18. The code editor window shows the C program '4_prime10.c' which includes a helper function 'checkPrime' to determine if a number is prime.

```
4_prime10.c
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)
8             c++;
9
10    if(c == 1)  return 1;
11    else        return 0;
12
13 }
14
15 int main()
16 {
17     int num, c=0;
18
19     printf("Enter the number : ");
20     scanf("%d", &num);
21
22     for(int i=1; i<=num; i++)
23     {
24         if(checkPrime(i))
25             printf("%d ", i);
26     }
27     printf("\n");
28
29 }
```

```
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 top right corner, there is a system tray icon for 'WB'. On the desktop, there are icons for 'WB' and 'New Folder'. In the bottom left corner, there is a dock with various application icons. The main focus is a terminal window titled 'Linux Lite Terminal' in the top right. The terminal shows the command 'g 5_find_output 5_find_output.c' followed by the output of the program execution: 'osboxes ~ Desktop New Folder ass_6 ./5_find_output' and '12'. Below the terminal is a code editor window titled '5_find_output.c'. The code is a C program that finds the maximum value in an array. It includes comments explaining the logic: //find max in array and //print max value from array. The code editor also shows a file tree on the left side.

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int i, a[4]={3, 1, 12, 4}, result;
    result=a[0];

    //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;
}
```

Solution 6

A screenshot of a Linux desktop environment, similar to the one in Solution 5. In the top right corner, there is a system tray icon for 'WB'. On the desktop, there are icons for 'WB' and 'New Folder'. In the bottom left corner, there is a dock with various application icons. The main focus is a terminal window titled 'Linux Lite Terminal' in the top right. The terminal shows the command 'g 6_find_output 6_find_output.c' followed by the output of the program execution: 'osboxes ~ Desktop New Folder ass_6 ./6_find_output' and 'i = 5, j = 5, k = 2'. Below the terminal is a code editor window titled '6_find_output.c'. The code is a C program that prints the values of three variables: i, j, and k. The code editor also shows a file tree on the left side.

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int arr[] = {1,2,3,4,5,6};
    int i, j, K;

    j = ++arr[2]; //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;
}
```

Solution 7

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 are icons for 'WB' and 'New Folder'. A file manager window titled 'final ass.docx - LibreOffice - ~/Desktop/New Folder...' is open, showing a list of files including 'ass_10', 'ass_2', 'ass_3', 'ass_4', 'ass_5', 'ass_6', '10_largement_arr.c', '10_largement_arr.c', '11_reverse_arr.c', '11_reverse_arr.c', '12_merge_arr.c', '12_merge_arr.c', '13_dup_del.c', '13_dup_del.c', '14_del_elem.c', '14_del_elem.c', '15_sum_ele.c', '15_sum_ele.c', '16_subtract.c', '16_subtract.c', '17_low_triangle.c', '17_low_triangle.c', '1 Armstrong.c', '1 Armstrong.c', '2 palindrome.c', '2 palindrome.c', '3 prime.c', '3 prime.c', '4 pyramid.c', '4 pyramid.c', '5 find_output.c', '5 find_output.c', '6 find_output.c', '6 find_output.c', '7 find_output.c', '7 find_output.c', '8 find_output.c', '8 find_output.c', and '9 find_output.c'. Below the file manager is a terminal window titled 'Linux Lite Terminal' with the command 'osboxes ~ Desktop New Folder ass_6 g 7_find_output 7_find_output.c' and the output 'IIT MADRAS'. Another terminal window is partially visible with the command 'osboxes ~ Desktop New Folder ass_6 ./7_find_output'.

```
#include <stdio.h>

int main()
{
    int a[6] = {1, 2, 3, 4, 5, 6};

    switch(sizeof(a)){ //return 24, no case matches
        case 1:
        case 2:
        case 3:
        case 4:
        case 5: printf("IIT KGP\n");
                  break;
    }

    // print IIT MADRAS
    printf("IIT MADRAS \n");
    return 0;
}
```

Solution 8

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 are icons for 'WB' and 'New Folder'. A file manager window titled 'final ass.docx - LibreOffice - ~/Desktop/New Folder...' is open, showing a list of files including 'ass_10', 'ass_2', 'ass_3', 'ass_4', 'ass_5', 'ass_6', '10_largement_arr.c', '10_largement_arr.c', '11_reverse_arr.c', '11_reverse_arr.c', '12_merge_arr.c', '12_merge_arr.c', '13_dup_del.c', '13_dup_del.c', '14_del_elem.c', '14_del_elem.c', '15_sum_ele.c', '15_sum_ele.c', '16_subtract.c', '16_subtract.c', '17_low_triangle.c', '17_low_triangle.c', '1 Armstrong.c', '1 Armstrong.c', '2 palindrome.c', '2 palindrome.c', '3 prime.c', '3 prime.c', '4 pyramid.c', '4 pyramid.c', '5 find_output.c', '5 find_output.c', '6 find_output.c', '6 find_output.c', '7 find_output.c', '7 find_output.c', '8 find_output.c', '8 find_output.c', and '9 find_output.c'. Below the file manager is a terminal window titled 'Linux Lite Terminal' with the command 'osboxes ~ Desktop New Folder ass_6 g 8_find_output 8_find_output.c' and the output 'a a a a a a'. Another terminal window is partially visible with the command 'osboxes ~ Desktop New Folder ass_6 ./8_find_output'.

```
#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;
}
```

Solution 9

A screenshot of a Linux desktop environment. In the top right corner, there is a system tray icon for 'WB'. On the desktop, there are icons for 'New Folder' and 'assignments'. In the bottom right corner, there is a dock with various application icons. The main focus is a terminal window titled 'Linux Lite Terminal' which shows the command:

```
osboxes ~ Desktop New Folder ass_6 g 9_find_output 9_find_output.c  
osboxes ~ Desktop New Folder ass_6 ./9_find_output  
4, 4  
osboxes ~ Desktop New Folder ass_6
```

Below the terminal is a code editor window titled 'Sublime Text (UNREGISTERED)'. It displays a C program named '9_find_output.c'. The code includes a main function that prints the value of arr[1]. The terminal output shows the result as 4.

```
#include <stdio.h>  
int main()  
{  
    int a;  
    int arr[5] = {1, 2, 3, 4, 5};  
    //arr[1]= ++2 = 3  
    arr[1] = ++arr[1];  
    //arr[1]= arr[3] = 4, a = 3++ = 4  
    arr[1] = arr[a++];  
    printf("%d\n", a, arr[1]); // a=4, arr[1]=4  
    return 0;  
}
```

Solution 10

A screenshot of a Linux desktop environment, similar to the one above. In the top right corner, there is a system tray icon for 'WB'. On the desktop, there are icons for 'New Folder' and 'assignments'. In the bottom right corner, there is a dock with various application icons. The main focus is a terminal window titled 'Linux Lite Terminal' which shows the command:

```
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
```

Below the terminal is a code editor window titled 'Sublime Text (UNREGISTERED)'. It displays a C program named '10_largement_arr.c'. The code reads elements from standard input and finds the maximum value. The terminal output shows the largest element as 200.

```
// Write a C Program to find Largest Element of an Integer Array.  
// Here the number of elements in the array 'n' and the elements of  
// the array is read from the test data.  
// Use the printf statement given below to print the largest element.  
// printf("Largest element = %d", largest);  
#include <stdio.h>  
int main()  
{  
    int t;  
    scanf("%d", &t);  
    int a[t], max=1000;  
    // printf("Enter the elements : ");  
    for (int i = 0; i < t; ++i)  
    {  
        scanf("%d", &a[i]);  
        if(max < a[i])  
            max = a[i];  
    }  
    printf("Largest element = %d\n", max);  
    return 0;  
}
```

Solution 11

```
11_reverse_arr.c
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--){
24         printf("%d ", a[t]);
25     }
26
27     printf("\n");
28     return 0;
29 }
```

```
osboxes ~ Desktop New Folder ass_6 g 11_reverse_arr 11_reverse_arr.c
osboxes ~ Desktop New Folder ass_6 ./11_reverse_arr
6
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

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

```
osboxes ~ Desktop New Folder ass_6 g 12_merge_arr 12_merge_arr.c
osboxes ~ Desktop New Folder 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 right corner, there is a system tray icon for 'WB'. On the desktop, there is a 'New Folder' icon. Two terminal windows are open in the bottom right corner. The left terminal window shows the output of a command to run a C program named '13_dup_del.c'. The right terminal window shows the execution of the program, which prints the numbers 6, 50, 6, 7, 7, 2, 7, 50, 6, 7, 2, 7. A code editor window titled '13_dup_del.c' is open in the center, displaying C code to delete duplicates from an array. The code includes comments explaining the logic: it first takes the size of the first array, then inputs all elements. It then loops through the array to delete duplicates by comparing each element with the rest of the array.

```
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]) { i++; }
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                 i++;
23             }
24
25         for (int i = 0; i < t; i++)
26             printf("%d", a[i]);
27     }
28
29     return 0;
30 }
```

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. Two terminal windows are open in the bottom right corner. The left terminal window shows the output of a command to run a C program named '14_del_elem.c'. The right terminal window shows the execution of the program, which prints the numbers 5, 9, 10, 11, 1, 20, 5, 9, 10, 11, 1. A code editor window titled '14_del_elem.c' is open in the center, displaying C code to delete an element from a specified location in an array. The code includes comments explaining the logic: it first takes the size of the array, then inputs all elements. It then asks for the position of the element to be deleted and performs the deletion by shifting elements.

```
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 }
```

Solution 15

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 right corner, there is a dock with various icons.

Code Editor Window:

```
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 // #include <stdio.h>
13 int main(){
14     int arr[3][3], sum[3]={0, 0, 0};
15     for (int i = 0; i < 3; i++)
16         for (int j = 0; j < 3; j++)
17             scanf("%d", &arr[i][j]);
18     sum[i] += arr[i][j];
19     printf ("\n");
20     for (int i = 0; i < 3; i++)
21         printf ("%d\n", sum[i]);
22     return 0;
23 }
24
25
26
27
28
29 }
```

Terminal Window:

```
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
```

Solution 16

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 right corner, there is a dock with various icons.

Code Editor Window:

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

Terminal Window:

```
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
```

Solution 17

The screenshot shows a Linux desktop environment with a terminal window titled "Linux Lite Terminal" and a code editor window titled "17_low_triangle.c".

The terminal window displays the following command and output:

```
osboxes ~ Desktop New Folder ass_6 g 17_low_triangle 17_low_triangle.c
osboxes ~ Desktop New Folder ass_6 ./17_low_triangle
2 3 4
5 6 0
4 5 6
```

The code editor window contains the C program "17_low_triangle.c" which prints the lower triangle of a square matrix. The code uses nested loops to iterate through the matrix elements and prints them if they are on or below the main diagonal.

```
#include <stdio.h>

int main(){
    int arr[3][3], k=1;
    for (int i = 0; i < 3; i++){
        for (int j = 0; j < i; j++)
            scanf("%d", &arr[i][j]);
        printf("\n");
        for (int i = 0; i < 3; i++){
            for (int j = 0; j < i; j++){
                if(j<k)
                    printf("%d ", arr[i][j]);
                else
                    printf("0 ");
            }
            printf("\n");
        }
    }
    return 0;
}
```

DAY 7

Solution 1

A screenshot of a Linux desktop environment. On the left, a code editor window titled "1_count_UL.c" shows a C program to count uppercase and lowercase letters. The terminal window on the right shows the output of running the program, which prints "Hello from CDAC hyderabad", followed by the counts of uppercase and lowercase letters.

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

```
osboxes ~ Desktop New Folder ass_7 g 1.count_UL 1_count_UL.c
osboxes ~ Desktop New Folder ass_7 ./1_count_UL
Hello from CDAC hyderabad
Uppercase Letters : 5
Lowercase Letters : 17
osboxes ~ Desktop New Folder ass_7
```

Solution 2

A screenshot of a Linux desktop environment. On the left, a code editor window titled "2_word_length.c" shows a C program to find the largest and smallest words in a sentence. The terminal window on the right shows the output of running the program, which prints the string "Problem solving by C programming", followed by the largest and smallest words.

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

```
osboxes ~ Desktop New Folder ass_7 g 2_word_length 2_word_length.c
osboxes ~ Desktop New Folder ass_7 ./2_word_length
Enter the String : Problem solving by C programming
Largest word is: programming
Smallest word is: C
osboxes ~ Desktop New Folder ass_7
```

Solution 3

The screenshot shows a Linux desktop environment with a dark theme. On the left is a code editor window titled "3_reverse.c" containing C code to reverse a string. On the right is a terminal window titled "Linux Lite Terminal" showing the execution of the program and its output.

```
3_reverse.c
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]s", d);
12
13    int l = strlen(d)-1;
14
15    printf("The string after reverse : ");
16    while(l >= 0){
17        printf("%c", d[l]);
18        l--;
19    }
20    printf("\n");
21
22    return 0;
23 }
```

```
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. On the left is a code editor window titled "4_prime_check.c" containing C code to check if a number is prime. On the right is a terminal window titled "Linux Lite Terminal" showing the execution of the program and its output.

```
4_prime_check.c
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/i == 0)
10             c++;
11
12     if(c == 1)  return 1;
13     else        return 0;
14 }
```

```
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

A screenshot of a Linux desktop environment. On the left, a Sublime Text window titled "5_power_rec.c" is open, displaying C code for calculating the power of a number using recursion. On the right, a terminal window titled "Linux Lite Terminal" shows the command `g 5_power_rec 5_power_rec.c` being run, followed by the output of the program which asks for a number and power, and then prints the result.

```
5_power_rec.c
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 }
```

```
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

A screenshot of a Linux desktop environment. On the left, a Sublime Text window titled "6_binary.c" is open, displaying C code for printing the binary equivalent of an integer using recursion. On the right, a terminal window titled "Linux Lite Terminal" shows the command `g 6_binary 6_binary.c` being run, followed by the output of the program which asks for a number and then prints its binary representation.

```
6_binary.c
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 }
```

```
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

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for 'WB'. Below it, a file manager window titled 'New Folder' is open, showing a list of files including 'ass_10', 'ass_2', 'ass_3', 'ass_4', 'ass_5', 'ass_6', 'ass_7', 'ass_value_1.c', 'ass_value_1.c', 'ass_11_find_output.c', 'ass_12_find_output.c', 'ass_13_return_value.c', 'ass_14_count_HI.c', 'ass_15_find_output.c', 'ass_16_find_output.c', 'ass_17_count_factoric.c', 'ass_18_count_factoric.c', 'ass_19_find_output.c', 'ass_20_find_output.c', 'ass_21_count_16.c', 'ass_22_count_16.c', 'ass_23_word_length.c', 'ass_24_word_length.c', 'ass_25_reverse.c', 'ass_26_reverse.c', 'ass_27_prime_check.c', 'ass_28_prime_check.c', 'ass_29_power_rec.c', 'ass_30_power_rec.c', 'ass_31_binary.c', 'ass_32_binary.c', 'ass_33_find_output.c', 'ass_34_find_output.c', 'ass_35_fill_blank.c', and 'ass_36_fill_blank.c'. A terminal window titled 'Linux Lite Terminal' is open, showing the command 'g 7_find_output 7_find_output.c' and its output: 'n1 = 18, n2 = 17'. The bottom of the screen features a dock with various application icons.

```
7_find_output.c
1 #include <stdio.h>
2
3 int main()
4 {
5     char str1[] = "Week-7-Assignment";
6     char str2[] = {'W', 'e', 'e', 'k', '-', '7', '-', 'A', 's', 's', 'i', 'l', 'g', ' ', 'm', ' ', 'e', ' ', 'n', ' ', 't'};
7
8     //str1 carries string which contain nullchar in the end 17+
9     //str2 is array of char ..size=no of char here 17
10    int n1 = sizeof(str1)/sizeof(str1[0]); // 18
11    int n2 = sizeof(str2)/sizeof(str2[0]); // 17
12
13    printf("n1 = %d, n2 = %d\n", n1, n2); //prints n1 = 18, n2 = 17
14
15    return 0;
16
17 }
```

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

Solution 8

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for 'WB'. Below it, a file manager window titled 'New Folder' is open, showing a list of files including 'ass_2', 'ass_3', 'ass_4', 'ass_5', 'ass_6', 'ass_7', 'ass_value_1.c', 'ass_value_1.c', 'ass_11_find_output.c', 'ass_12_find_output.c', 'ass_13_return_value.c', 'ass_14_count_HI.c', 'ass_15_find_output.c', 'ass_16_find_output.c', 'ass_17_count_factoric.c', 'ass_18_count_factoric.c', 'ass_19_find_output.c', 'ass_20_find_output.c', 'ass_21_count_16.c', 'ass_22_count_16.c', 'ass_23_word_length.c', 'ass_24_word_length.c', 'ass_25_reverse.c', 'ass_26_reverse.c', 'ass_27_prime_check.c', 'ass_28_prime_check.c', 'ass_29_power_rec.c', 'ass_30_power_rec.c', 'ass_31_binary.c', 'ass_32_binary.c', 'ass_33_find_output.c', 'ass_34_find_output.c', 'ass_35_fill_blank.c', and 'ass_36_fill_blank.c'. A terminal window titled 'Linux Lite Terminal' is open, showing the command 'g 8_find_output 8_find_output.c' and its output: 'hello world'. The bottom of the screen features a dock with various application icons.

```
8_find_output.c
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     char str1[20] = "hello", str2[20] = " world";
7
8     // strcat - concat both var string into str1
9     // strcpy - copy 2nd arg string into str2
10
11    printf("%s\n", strcpy(str2, strcat(str1, str2)));
12
13    //str1 = "hello world" and string copy to var str2 and then it prints "hello world"
14
15    return 0;
16
17 }
```

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

Solution 9

A screenshot of a Linux desktop environment. In the top right corner, there is a system tray icon for 'WB'. Below it, a 'New Folder' icon is visible. The main window is a terminal titled 'Linux Lite Terminal' with the command 'osboxes ~ Desktop New Folder ass_7 g 9_fill_blank 9_fill_blank.c ./9_fill_blank' and the output 'Hwi orea'. To the left of the terminal is a code editor window titled 'final ass docx - LibreOffice - /Desktop/New Folder assignments - Sublime Text (UNREGISTERED)'. The file '9_fill_blank.c' contains the following C code:

```
#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

A screenshot of a Linux desktop environment. In the top right corner, there is a system tray icon for 'WB'. Below it, a 'New Folder' icon is visible. The main window is a terminal titled 'Linux Lite Terminal' with the command 'osboxes ~ Desktop New Folder ass_7 g 10_value_i 10_value_i.c ./10_value_i' and the output 'daffodills'. To the left of the terminal is a code editor window titled 'final ass docx - LibreOffice - /Desktop/New Folder assignments - Sublime Text (UNREGISTERED)'. The file '10_value_i.c' contains the following C code:

```
#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 then 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

A screenshot of a Linux desktop environment. In the top right corner, there is a system tray icon for 'WB'. On the desktop, there are icons for 'WB' and 'New Folder'. In the bottom right corner, there is a 'Linux Lite Terminal' window. The terminal window shows the command 'osboxes ~ Desktop New Folder ass_7 g 11_find_output 11_find_output.c' and its output 'osboxes ~ Desktop New Folder ass_7 ./11_find_output'. Below the terminal, the desktop environment includes a dock with various application icons.

```
11_find_output.c
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     char p[] = "assignment";
7     char t;
8     int i, j;
9
10    // swapping of char array or reversing array
11    for (int i = 0, j = strlen(p); i<j ; i++)
12    {
13        t = p[i];
14        p[i] = p[j-i];
15        p[j-i] = t;
16    }
17
18    // '|0' or null char use to represent end of the character string,
19    // during the reversing elements of array the '|0' null character comes at first
20    // location p[0] = '|0' and '|0' also shows the termination of string
21    // that is the reason the it prints nothing on output console
22
23    printf("%s", p);
24
25    return 0;
26 }
```

Solution 12

A screenshot of a Linux desktop environment. In the top right corner, there is a system tray icon for 'WB'. On the desktop, there are icons for 'WB' and 'New Folder'. In the bottom right corner, there is a 'Linux Lite Terminal' window. The terminal window shows the command 'osboxes ~ Desktop New Folder ass_7 g 12_find_output 12_find_output.c' and its output 'osboxes ~ Desktop New Folder ass_7 ./12_find_output' followed by the numbers '67, 67'. Below the terminal, the desktop environment includes a dock with various application icons.

```
12_find_output.c
1 #include <stdio.h>
2 void fu(int x, int y);
3
4 int main(int argc, char const *argv[])
5 {
6     char x=67, y='C'; // ASCII value of C = 67 assign to var y
7     fu(x, y); // calling fu with passing values(67, 67)
8
9     return 0;
10 }
11 void fu(int x, int y){ //print 67, 67
12     printf("%d, %d\n", x, y);
13 }
```

Solution 13

A screenshot of a Linux desktop environment. At the top, there's a window titled "final ass docx - LibreOffice Writer". Below it is a terminal window titled "Linux Lite Terminal" with the command "osboxes@osboxes: ~ Desktop New Folder ass_7 g 13_return_value 13_return_value.c" and the output "osboxes@osboxes: ~ Desktop New Folder ass_7 ./13_return_value". In the center, a Sublime Text window titled "13_return_value.c" is open, displaying the following C code:

```
1 #include <stdio.h>
2
3 // return number 5 with input 5 or less than 5
4 // return same passing number if greater than 5
5 int f(int n){
6     static int i=1;
7     if(n>=5)
8         return n;
9     n = n + 1;
10    i++;
11
12    return f(n);
13}
14
15 int main(){
16     printf("%d\n", f(10)); // calling function f and print return value from f
17     return 0;
18 }
```

The file manager on the left shows a folder structure with files like "ass_5", "ass_6", "ass_7", etc. The bottom taskbar has icons for various applications.

Solution 14

A screenshot of a Linux desktop environment. At the top, there's a window titled "final ass docx - LibreOffice Writer". Below it is a terminal window titled "Linux Lite Terminal" with the command "osboxes@osboxes: ~ Desktop New Folder ass_7 g 14_count_Hi 14_count_Hi.c" and the output "osboxes@osboxes: ~ Desktop New Folder ass_7 ./14_count_Hi" followed by "Hello". In the center, a Sublime Text window titled "14_count_Hi.c" is open, displaying the following C code:

```
1 #include <stdio.h>
2
3 int i; //i=0
4 int fun();
5
6 int main()
7 {
8     while(i){ //0=false i.e while never start
9         //never get call
10        fun();
11        main();
12    }
13    printf("Hello\n"); //print only Hello
14
15    return 0;
16 }
17
18 int fun(){
19     printf("Hi");
20 }
```

The file manager on the left shows a folder structure with files like "ass_5", "ass_6", "ass_7", etc. The bottom taskbar has icons for various applications.

Solution 15

A screenshot of a Linux desktop environment. 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 right corner, there is a dock with various icons. A terminal window titled 'Linux Lite Terminal' is open, showing the command:

```
osboxes ~ Desktop New Folder ass_7 g 15_find_output 15_find_output
27.08 osboxes ~ Desktop New Folder ass_7 ./15_find_output
```

To the left of the terminal, a code editor window titled '15_find_output.c' is open, displaying the following C code:

```
#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("%0.2f\n", result); //print result avg of age = 27.00
    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

A screenshot of a Linux desktop environment. 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 right corner, there is a dock with various icons. A terminal window titled 'Linux Lite Terminal' is open, showing the command:

```
osboxes ~ Desktop New Folder ass_7 g 16_find_output 16_find_output
10 osboxes ~ Desktop New Folder ass_7 ./16_find_output
```

To the left of the terminal, a code editor window titled '16_find_output.c' is open, displaying the following C code:

```
#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++;
    }
    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, there is a file manager window showing a directory structure with files like 'ass_5', 'ass_6', 'ass_7', etc. In the center, there is a Sublime Text editor window displaying a C program named '17_count_factexc.c'. The code defines a factorial function and prints its value. On the right, there is a terminal window titled 'Linux Lite Terminal' showing the command 'g 17_count_factexc 17_count_factexc.c' being run, followed by the output '10! = 3628800'. The desktop background is a colorful gradient.

```
17_count_factexc.c
1 #include <stdio.h>
2
3 int factorial(int);
4
5 int main()
6 {
7     int n=10;
8     long int f;
9
10    f = factorial(n); //Function calling
11    printf("%d! = %ld\n", n, f);
12
13    return 0;
14}
15 // factorial func is called only one times which means fuctions executed one times only.
16 int factorial(int n){
17    if(n==0)
18        return 1;
19    else
20        return n*factorial(n-1);
21}
22 }
```

```
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, there is a file manager window showing a directory structure with files like 'ass_5', 'ass_6', 'ass_7', etc. In the center, there is a Sublime Text editor window displaying a C program named '18_find_output.c'. The code swaps two integers and prints them before and after the swap. On the right, there is a terminal window titled 'Linux Lite Terminal' showing the command 'g 18_find_output 18_find_output.c' being run, followed by the output 'Before swapping num1 = 10 num2 = 20' and 'After swapping num1 = 10 num2 = 20'. The desktop background is a colorful gradient.

```
18_find_output.c
1 #include <stdio.h>
2
3 void swap(int a, int b){
4     int temp;
5     temp = a;
6     a = b;
7     b = temp;
8 }
9
10 int main()
11 {
12     int num1 = 10, num2 = 20;
13
14     printf("Before swapping num1 = %d num2 = %d\n", num1, num2);
15     swap(num1, num2);
16     printf("After swapping num1 = %d num2 = %d\n", num1, num2);
17
18     return 0;
19 }
20
21 // output
22
23 // Before swapping num1 = 10 num2 = 20
24 // After swapping num1 = 10 num2 = 20
25 // swapping not happened because here value passing with call by value not using call by reference.
26 // 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 several windows open:

- File Manager:** Shows a tree view of files and folders. The current folder is `Desktop/New Folder`. Visible files include `ass_5`, `ass_6`, `ass_7`, `10_value_1.c`, `10_value_1.c`, `11_find_output.c`, `11_find_output.c`, `12_find_output.c`, `12_find_output.c`, `13_return_value.c`, `13_return_value.c`, `14_count_H.c`, `14_count_H.c`, `15_find_output.c`, `15_find_output.c`, `16_find_output.c`, `16_find_output.c`, `17_count_Factexc.c`, `17_count_Factexc.c`, `18_find_output.c`, `18_find_output.c`, `19_find.c`, `19_find.c`, `1.count_16.c`, `1.count_16.c`, `2.word_length.c`, `2.word_length.c`, `3.reverse.c`, `3.reverse.c`, `4.prime_check.c`, `4.prime_check.c`, `5.power_rec.c`, `5.power_rec.c`, `6.binary.c`, `6.binary.c`, `7.find_output.c`, `7.find_output.c`, `8.find_output.c`, `8.find_output.c`, `9.fill_blank.c`, `9.fill_blank.c`, and `tmpCodeHumanFile.c`.
- Code Editor:** Sublime Text window titled `19_find.c` containing C code for finding the middle value between three parameters.
- Terminal:** Linux Lite Terminal window showing the command `g 19_find 19_find.c` being run, followed by the output `./19_find`.

The desktop background is a colorful gradient, and the taskbar at the bottom shows various application icons.

```
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);
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 }
```

Line 17, Column 69
part assignments

DAY 8

Solution 1

A screenshot of a Linux desktop environment. On the left, a code editor window titled 'final ass docx - LibreOffice - ~/Desktop/New Folder...' shows a C program named '1_multi.c'. The code implements multiplication using recursion. On the right, a terminal window titled 'Linux Lite Terminal' shows the command './1_multi' being run, followed by the input '5 3' and the output '5 * 3 = 15'. The desktop background is a colorful gradient, and there are icons for 'WB' and 'New Folder' on the right.

```
// Write a program to implement multiplication using addition. Use recursion.  
#include <stdio.h>  
int multi(int a, int b){ // 2  
    static int sum = 0;  
    if(b == 0)  
        return sum;  
    else{  
        sum += a;  
        multi(a, b-1);  
    }  
}  
int main(){  
    int num1, num2;  
    printf("Enter the number : ");  
    scanf("%d%d", &num1, &num2);  
    printf("%d * %d = %d\n", num1, num2, multi(num1, num2));  
    return 0;  
}
```

Solution 2

A screenshot of a Linux desktop environment. On the left, a code editor window titled 'final ass docx - LibreOffice - ~/Desktop/New Folder...' shows a C program named '2_swap.c'. The code swaps two integers using a temporary variable. On the right, a terminal window titled 'Linux Lite Terminal' shows the command './2_swap' being run, followed by the input '6 9' and the output 'Before swapping : a = 6, b = 9' and 'After swapping : a = 9, b = 6'. The desktop background is a colorful gradient, and there are icons for 'WB' and 'New Folder' on the right.

```
// Write a program to swap two numbers.  
#include <stdio.h>  
int swap(int *a, int *b){  
    int t = *a;  
    *a = *b;  
    *b = t;  
}  
int main(){  
    int a, b;  
    printf("Enter the two numbers : ");  
    scanf("%d%d", &a, &b);  
    printf("Before swapping : a = %d, b = %d\n", a, b);  
    swap(&a, &b);  
    printf("After swapping : a = %d, b = %d\n", a, b);  
    return 0;  
}
```

Solution 3

A screenshot of a Linux desktop environment. On the left, a Sublime Text window titled 'final ass docx - LibreOffice - ~/Desktop/New Folder - Sublime Text (UNREGISTERED)' shows a C program named '3_sum.c'. The code defines a recursive function 'sum' to calculate the sum of integers. On the right, a terminal window titled 'Linux Lite Terminal' shows the command 'g 3-sum 3-sum.c' being run, followed by the output 'Enter the number : 5' and 'Sum of integers : 15'. The desktop background is a colorful gradient.

```
#include <stdio.h>

int sum(int num){
    static int s=0;
    if(num==0)
        return s;
    else
    {
        s+=num;
        sum(num-1);
    }
}

int main(){
    int num;
    printf("Enter the number : ");
    scanf("%d", &num);
    printf("Sum of integers : %d\n", sum(num));
    return 0;
}
```

Solution 4

A screenshot of a Linux desktop environment. On the left, a Sublime Text window titled 'final ass docx - LibreOffice - ~/Desktop/New Folder - Sublime Text (UNREGISTERED)' shows a C program named '4_concat.c'. The code concatenates two strings, 'Hello' and ' world', without using library functions. On the right, a terminal window titled 'Linux Lite Terminal' shows the command 'g 4-concat 4_concat.c' being run, followed by the output 'Hello world'. The desktop background is a colorful gradient.

```
#include <stdio.h>

int main(){
    char a[] = "Hello", b[] = " world";
    char s[40];
    for (int i = 0, j = 0, k = 0; j<sizeof(b); ++i)
    {
        if(a[i] == '\0')
            continue;
        if(i < sizeof(a))
            s[k] = a[i], k++;
        else
            s[k] = b[j], j++, k++;
    }
    printf("%s\n", s);
    return 0;
}
```

Solution 5

A screenshot of a Linux desktop environment. On the left, there is a Sublime Text window titled '(Desktop/New Folder) ass_8 - Sublime Text (UNREGISTERED)'. It contains a C program named '5_copy_string.c' which copies a string without using library functions. On the right, there is a terminal window titled 'Linux Lite Terminal' with the command 'g 5_copy_string 5_copy_string.c' and the output 'Welcome to Future'.

```
// Write a c program to copy string without using library functions
#include <stdio.h>
void copy(char a[], char s[], int n){
    for (int i = 0; i < n; ++i)
        s[i] = a[i];
}
int main(){
    char a[] = "Welcome to Future", s[10];
    copy(a, s, sizeof(a));
    printf("%s\n", s);
    return 0;
}
```

Solution 6

A screenshot of a Linux desktop environment. On the left, there is a Sublime Text window titled '(Desktop/New Folder) ass_8 - Sublime Text (UNREGISTERED)'. It contains a C program named '6_string_len.c' which finds the length of a string without using library functions. On the right, there is a terminal window titled 'Linux Lite Terminal' with the command 'g 6_string_len 6_string_len.c' and the output 'String : Programmer' and 'size of string : 10'.

```
// Write a c program to find length of string without using library functions
#include <stdio.h>
int main(){
    char a[] = "Programmer";
    int size=0;
    for (int i = 0; a[i] != '\0'; ++i)
        size++;
    printf("String : %s\n", a);
    printf("size of string : %d\n", size);
    return 0;
}
```

DAY 9

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are icons for LibreOffice, File Manager, and a terminal window titled "Linux Lite Terminal". The desktop background is a colorful abstract image.

File Manager: Shows a directory structure under "New Folder". One file, "1_pow.c", is open in Sublime Text. The code calculates a power using the `pow()` function from `math.h`.

```
// Write a C program to calculate power a given number using pow(a,b) function in math.h with evaluates: a^b
// Use #include<math.h> in your c program.

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

int power(int number, int pow){
    if(pow == 1)
        return number;
    else
        return number * power(number, pow-1);
}

int main()
{
    double num, pw;
    printf("Enter the number : ");
    scanf("%lf", &num, &pw);
    printf("%f\n", pow(num, pw));
    return 0;
}
```

Terminal: Shows the terminal window with the same title. It displays the compilation of "1_pow.c" using gcc and the execution of the resulting binary "1_pow". The user enters "2 5" and gets output "25.00". Then they enter "3 0.5" and get output "1.73".

```
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/cc4mtExw.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_g_1_pow 1_pow.c -lms
osboxes -> Desktop > New Folder > ass_9 > ./1_pow
Enter the number : 2 5
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 icons for LibreOffice, File Manager, and a terminal window titled "Linux Lite Terminal". The desktop background is a colorful abstract image.

File Manager: Shows a directory structure under "New Folder". One file, "2_leap_yr.c", is open in Sublime Text. The code implements a function to check if a year is a leap year.

```
// Write a program and implement a function to find if it is leap year or not.
// * is_leap_year() returns 1 if leap year
// * is_leap_year() returns 0 if NOT leap year
// Take 4 digit year as input for example: 2000

#include <stdio.h>

int is_leap_year(int year){
    if(year%400==0 || year%100 != 0 && year%4==0)
        return 1;
    else
        return 0;
}

int main(){
    int year;
    printf("Enter the year : ");
    scanf("%d", &year);

    if(is_leap_year(year))
        printf("\n%d is a leap year \n", year);
    else
        printf("\n%d is not a leap year \n", year);
    return 0;
}
```

Terminal: Shows the terminal window with the same title. It displays the compilation of "2_leap_yr.c" using gcc and the execution of the resulting binary "2_leap_yr". The user enters "2000" and gets output "2000 is a leap year". Then they enter "2012" and get output "2012 is a leap year". Finally, they enter "2022" and get output "2022 is not a leap year".

```
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

The screenshot shows a Linux desktop environment with several windows open. In the foreground, there is a terminal window titled "Linux Lite Terminal" showing the output of a factorial program. The terminal window has a dark background with light-colored text. Below the terminal is a dock bar with various icons. In the background, there is a file manager window titled "final ass dock - LibreOffice - ~Desktop/New Folder... [assignments - File Manager - Linux Lite Terminal]" showing a file tree with several C source files. The desktop background is a colorful abstract image.

```
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
10    printf("Enter the number : ");
11    scanf("%ld", &num);
12
13    printf("factorial(%ld) : %ld\n", num, factorial(num));
14
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 }
```

```
osboxes ~ Desktop/New Folder/ass_9 g 3_fact 3_fact.c
osboxes ~ Desktop/New Folder/ass_9 ./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
```

Solution 4

Solution 5

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

- File Manager:** Shows a tree view of files and folders, including subfolders for assignments 1 through 9.
- Terminal:** A "Linux Lite Terminal" window showing the command line interface.
- Running Application:** A window titled "5_exp.c" containing the C code for calculating the exponential value of a number using the `exp` function from `math.h`.

Code Content (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 }
```

Terminal Output:

```
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 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 terminal window titled "Linux Lite Terminal" and a file manager window titled "Sublime Text (HTML (GISTED))."

The terminal window displays the following output:

```
is present at location : 1
is present at location : 4
is present at location : 6
is present at location : 7
is present 4 times in an array
is not present in array
```

The file manager window shows a directory structure with files like `ass_10.c`, `ass_2.c`, `ass_3.c`, etc., and a file named `6_arr_loc.c`.

The code in `6_arr_loc.c` is as follows:

```
#include <stdio.h>

int main(){
    int t; scanf("%d", &t);
    int arr[t];
    for (int i = 0; i < t; i++)
        scanf("%d", &arr[i]);
    int search, count=0;
    scanf("%d", &search);
    for (int i = 0; i < t; i++)
        if(search == arr[i]){
            printf("%d is present at location : %d\n", search, i+1);
            count++;
        }
    if(count == 0)
        printf("%d is not present in array \n", search);
    else
        printf("%d is present %d times in an array\n", search, count);
    return 0;
}
```

Solution 7

The screenshot shows a Linux desktop environment with a terminal window titled "Linux Lite Terminal" and a code editor window titled "Sublime Text (UNREGISTERED)".

Terminal Window:

```
osboxes ~ Desktop New Folder ass_9 g 7_L_search 7_L_search.c
osboxes ~ Desktop New Folder ass_9 ./7_L_search
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
7
6

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

Code Editor Window:

```
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
11    for (int i = 0; i < t; i++)
12        scanf("%d", &arr[i]);
13
14    int search;
15    scanf("%d", &search);
16
17    for (int i = 0; i < t; i++)
18        if(search == arr[i])
19            printf("\n%d is present at location : %d\n", search, i+1), exit(0);
20
21    printf("\n%d is not present in the array\n", search);
22
23    return 0;
}
```

The terminal window shows the execution of the C program. It first asks for the size of the array (t) and the elements (arr). Then, it asks for the search value (search). Finally, it prints the result: "6 is present at location : 1".

Solution 8

The screenshot shows a Linux desktop environment with several windows open. In the foreground, a terminal window titled "Linux Lite Terminal" is active, displaying command-line output for reversing an array. The terminal shows three runs of the program, each printing the original array, the reversed array, and a final message. The code being run is a C program named "8_reverse_arr.c".

```
// Write a C program to reverse an array by using another new array
#include <stdio.h>
int main(){
    int t; scanf("%d", &t);
    int arr[t], rev_arr[t];
    for (int i = 0; i < t; i++)
        scanf("%d", &arr[i]);
    for (int i = 0, k = t-1; i < t; i++, k--)
        rev_arr[i] = arr[k];
    printf("\nOriginal array elements are : \n");
    for (int i = 0; i < t; i++)
        printf("%d ", arr[i]);
    printf("\n\nReversed array elements are : \n");
    for (int i = 0; i < t; i++)
        printf("%d ", rev_arr[i]);
    printf("\n");
    return 0;
}
```

Terminal Output:

```
osboxes ~ % Desktop > New Folder > ass_9 g 8_reverse_arr 8_reverse_arr.c
osboxes ~ % Desktop > New Folder > ass_9 ./8_reverse_arr
7
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_reverse_arr
5
50
60
40
30
20
Original array elements are :
50 60 40 30 20
Reversed array elements are :
20 30 40 60 50
osboxes ~ % Desktop > New Folder > ass_9
```

Solution 9

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a file manager, a terminal, and system status indicators. Below the panel, there is a dock with various application icons.

Sublime Text Window:

- File: 9_reverse_arr.c, 9_swap_arr.c
- Line 21, Column 11 - Field 3 of 3
- Spaces: 4

Code (9_swap_arr.c):

```
// Write a C program to reverse an array by swapping the elements and without using any new array
#include <stdio.h>

int main(){
    int t; scanf("%d", &t);
    int arr[t];

    for (int i = 0; i < t; i++)
        scanf("%d", &arr[i]);

    for (int i = 0, k = t-1; i < t/2; ++i, --k) {
        int t = arr[i];
        arr[i] = arr[k];
        arr[k] = t;
    }

    printf("\nReversed array elements are : \n");
    for (int i = 0; i < t; i++)
        printf("%d ", arr[i]);
}

printf("\n");
return 0;
}
```

Terminal Window:

```
osboxes ~ Desktop > New Folder > ass_9 g 9_swap_arr 9_swap_arr.c
osboxes ~ Desktop > New Folder > ass_9 ./9_swap_arr
5
50
60
40
30
20
Reversed array elements are :
20 30 40 60 50
osboxes ~ Desktop > New Folder > ass_9
```

DAY 10

Structures and Unions

Solution 1

The screenshot shows a Linux desktop environment with a terminal window titled "Linux Lite Terminal" and a code editor window titled "final ass docx - LibreOffice". The terminal window displays the output of a C program named "1_std_detail.c". The code defines a structure "std" with fields "id", "gender", and "name". It prompts the user to enter student ID, gender, and name, then prints them back. The terminal shows the input "Enter student ID : 101", "Student ID : 101", "Enter student gender : M", "Student gender : M", and "Enter student name : Itachi", "Student name : Itachi". The code editor window shows the source code for "1_std_detail.c".

```
#include <stdio.h>
struct std{
    int id;
    char gender;
    char name[20];
};
int main(){
    struct std stdl;
    printf("Enter the student ID : ");
    scanf("%d", &stdl.id);
    printf("Enter student gender : ");
    scanf(" %c", &stdl.gender);
    printf("Enter student name : ");
    scanf(" %[^\n]", stdl.name);
    printf("\nStudent ID : %d\n", stdl.id);
    printf("Student gender : %c\n", stdl.gender);
    printf("Student name : %s\n", stdl.name);
    return 0;
}
```

Solution 2

The screenshot shows a Linux desktop environment with a terminal window titled "Linux Lite Terminal" and a code editor window titled "final ass docx - LibreOffice". The terminal window displays the output of a C program named "2_personal.c". The code defines a structure "personal" with fields "name", "date_of_joining", and "salary". It prompts the user to enter information for 5 people and prints it back. The terminal shows the input for 5 people: name, date of joining, and salary. The code editor window shows the source code for "2_personal.c".

```
#include <stdio.h>
struct personal{
    char name[20];
    char date_of_joining[20];
    int salary;
};
int main(){
    struct personal person[5];
    for (int i = 0; i < 5; ++i){
        printf("\nEnter person name %d : ", i+1);
        scanf(" %[^\n]", person[i].name);
        printf("Enter date of joining : ");
        scanf(" %[^\n]", person[i].date_of_joining);
        printf("Enter person's salary : ");
        scanf("%d", &person[i].salary);
    }
    printf("\n-----Details-----");
    for (int i = 0; i < 5; ++i){
        printf("\nPerson name : %s\n", person[i].name);
        printf("Date of joining : %s\n", person[i].date_of_joining);
        printf("Person's salary : %d\n", person[i].salary);
    }
    return 0;
}
```

Solution 3

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Terminal Window:

```
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
osboxes -> New Folder > ass_10 > struct
```

Code Editor (Linux Lite Terminal -)

```
#include <stdio.h>
struct time_struct{
    int hour, minute, second;
};
int main(){
    struct time_struct time;
    printf("Enter Hours : ");
    scanf("%d", &time.hour);
    printf("Enter minute : ");
    scanf("%d", &time.minute);
    printf("Enter second : ");
    scanf("%d", &time.second);
    printf("\n Time : %d:%d:%d\n", time.hour, time.minute, time.second);
    return 0;
}
```

The code defines a structure `time_struct` with members `hour`, `minute`, and `second`. The `main` function prompts the user for hours, minutes, and seconds, then prints the time in the format `hh:mm:ss`.

Solution 4

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Terminal Window:

```
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
osboxes -> New Folder > ass_10 > struct
```

Code Editor (Linux Lite Terminal -)

```
#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-----");
    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;
}
```

The code defines a structure `cricket` with members `player_name`, `team_name`, and `batting_avg`. The `main` function reads three players and their details (name, team, and batting average) into an array, then prints them back out.

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 has tabs labeled "osboxes" and "New Folder > ass_10 > struct". The terminal output shows the execution of a C program named "5_std_record.c" which reads student data from a file and prints it to the screen.

```
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 studnet marks : ");
22         scanf("%f", &students_data[i].std_marks);
23
24     printf("\n-----Details-----\n");
25     for (int i = 0; i < 3; ++i)
26     {
27         printf("Student name : %s\n", students_data[i].std_name);
28         printf("Branch name : %s\n", students_data[i].std_branch);
29         printf("Studnet marks : %.2f\n", students_data[i].std_marks);
30     }
31
32
33     return 0;
34 }
```

```
osboxes - 2 New Folder > ass_10 > struct g 5_std_record 5_std_record.c
osboxes - 2 New Folder > ass_10 > struct ./5_std_record

Enter student name : Rahul
Enter branch name : EE
Enter studnet marks : 74.2

Enter student name : David
Enter branch name : CSE
Enter studnet marks : 87.9

Enter student name : Julie
Enter branch name : CSE
Enter studnet marks : 92

-----Details-----

Student name : Rahul
Branch name : EE
Studnet marks : 74.20

Student name : David
Branch name : CSE
Studnet marks : 87.90

Student name : Julie
Branch name : CSE
Studnet marks : 92.00
osboxes - 2 New Folder > ass_10 > struct
```

Preprocessor directives

Solution 1

The screenshot shows a Linux desktop environment with a terminal window titled "Linux Lite Terminal" and a code editor window titled "final ass docx - LibreOffice - assignments - File Manager". The terminal window displays the output of a C program that prints the value of PI and the speed of light. The code editor window shows a file named "1_header_var.c" containing C code that includes a header file "custom.h" and defines variables 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 #include <stdio.h>
4 #include "custom.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
14     return 0;
15 }
16
17 // Output : The PI value is 3.142857
18
```

```
osboxes: ~ New Folder > ass_10 > struct > cd ..>preprocessor directives> g 1_header_var 1_
header_var.c
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
osboxes: ~ New Folder > ass_10 > preprocessor directives >
```

Solution 2

The screenshot shows a Linux desktop environment with a terminal window titled "Linux Lite Terminal" and a code editor window titled "final ass docx - LibreOffice - assignments - File Manager". The terminal window displays the output of a C program that calculates the area of a circle given its radius. The code editor window shows a file named "2_define.c" containing C code that uses a preprocessor directive #define to declare the value of PI.

```
1 // Using #define, declare a value, say PI = 3.14, and use it to calculate area of a circle,
2 // given the radius.
3 #include <stdio.h>
4 #define PI 3.14
5
6 int main(){
7     int radius;
8
9     printf("Enter the radius : ");
10    scanf("%d", &radius);
11
12    printf("\nArea of circle : %.3f\n", PI*radius*radius);
13
14    return 0;
15 }
```

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

Solution 3

```
4 cricket.c      5 std_record.c    2 define.c      3 loop_macro.c
1 // Write a for loop to print numbers from 1 to 20. The for loops has to be written as a
2 // multiline macro
3
4 #include <stdio.h>
5 #define forlo(x, y)
6   {
7     for(int i=x; i<=y; ++i) \
8       printf("%d\n", i);
9     printf("\n");
10  }
11
12 int main(){
13   forlo(1, 20)
14   return 0;
15 }

osboxes... New Folder ass_10 preprocessor directives g 3_loop_macro 3_
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
loop_macro.c:5:24: warning: backslash and newline separated by space
5 | #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

```
4 cricket.c      5 std_record.c    2 define.c      3 loop_macro.c      4 ifdef.c
1 // Give example usage of conditional compilation using #ifdef, #ifndef and #undef.
2 #include <stdio.h>
3 #define VAR
4 #define VAR1 3
5
6 int main()
7 {
8   //return true if this macro is defined
9   #ifdef VAR
10   printf("var is defined\n");
11   #else
12   printf("var is not defined\n");
13   #endif
14
15   //undefines a preprocessor macro
16   #undef VAR
17
18   #ifdef VAR
19   printf("var is defined\n");
20   #else
21   printf("var is undefined\n");
22   #endif
23
24   //return true if this macro is not defined
25   #ifndef VAR1
26   printf("var1 is not defined\n");
27   #else
28   printf("var1 is defined\n");
29   #endif
30
31   return 0;
32 }
```

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

Solution 5

The screenshot shows a Linux desktop environment with the following details:

- File Manager:** Shows a folder structure under "New Folder" named "ass_10". Inside are files: 4_cricket.c, 5_std_records.c, 2_define.c, 3_loop_macro.c, and 5_concat.c.
- Terminal:** An "osboxes" terminal window titled "Linux Lite Terminal" is open. It displays the following command and output:

```
osboxes ~ New Folder > ass_10 preprocessors directives g 5_concat 5_concat.c
osboxes ~ New Folder > ass_10 preprocessors directives ./5_concat
123456
osboxes ~ New Folder > ass_10 preprocessors directives
```
- Taskbar:** The taskbar at the bottom shows icons for "pdf" and "assignments".

Storage Class Specifier

Solution 1

The screenshot shows a Linux desktop environment with a terminal window and a code editor window. The terminal window is titled 'Linux Lite Terminal' and contains the command: 'osboxes ~ > New Folder > ass_10 > storage class specifier g !_static !_static.c ./!_static'. The code editor window is titled 'final ass.docx - LibreOffice - ~/Desktop/New Folder...' and displays a C program named '1_static.c'. The code uses the 'static' keyword to declare a global static variable 'count' and a function 'fun()' that increments it. The main() function calls 'fun()' three times and prints its value.

```
1 // Using Static keyword, keep a count of the number of times a function was called.
2
3 #include <stdio.h>
4
5 //global static variable of int type
6 static int count=0;
7
8 //function return count of its called
9 int fun(){
10     return ++count;
11 }
12
13 int main(){
14     //calling fun function 3 times
15     fun();
16     fun();
17     fun();
18
19     printf("\n%d\n", count);
20
21     return 0;
22 }
```

Solution 2

The screenshot shows a Linux desktop environment with a terminal window and a code editor window. The terminal window is titled 'Linux Lite Terminal' and contains the command: 'osboxes ~ > New Folder > ass_10 > storage class specifier g bin 2_extern.c 2_extern2.c'. The code editor window is titled 'final ass.docx - LibreOffice - ~/Desktop/New Folder...' and displays two C files: '1_static.c' and '2_extern.c'. The '2_extern.c' file contains an 'extern int var;' declaration. The '1_static.c' file includes this file and prints the value of 'var'. A warning message is shown in the terminal regarding the redeclaration of 'var'.

```
1 // Using extern keywords, declare variable in another file(second.c) , and in another
2 // file(first.c), utilize that variable.
3
4 // 2.extern2.c contains
5 // extern int var = 420;
6 // var already declared in other file so that gives warning
7
8 #include <stdio.h>
9 extern int var;
10
11 int main(){
12
13     printf("var : %d\n", var);
14
15     return 0;
16
17 }
```

Solution 3

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Left Window):

```
1 static.c      2 extern.c      3_localvar.c      4_extend.c
 1 // Can two functions have variables with the same name?
 2 // Give example of they both have different scopes.
 3 // Yes, two or more function can have variable with the same name,
 4 // because that same name variable always to local to it specific function.
 5 // Which means that var inside func does exist or scope outside the func.
 6
 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 }
```

Terminal (Right Window):

```
osboxes ~ New Folder ass_10 storage class specifier 3_localvar 3_localvar.c
osboxes ~ New Folder ass_10 storage class specifier ./3_localvar
fun1 -> var = 4
fun2 -> var = 9
osboxes ~ New Folder ass_10 storage class specifier
```

The terminal output shows the execution of the program, printing the values of the variables 'var' from both functions.

File handling

Solution 1

The screenshot shows a Linux desktop environment with several windows open. In the top left, there's a LibreOffice Calc window titled 'final ass docx - LibreOffice'. Next to it is a terminal window titled 'Linux Lite Terminal' with the command 'cd ~/Desktop/New Folder...'. In the center, there's another terminal window titled 'Linux Lite Terminal' with the command 'g 1_write 1_write.c'. Below these are two file manager windows. The left one is titled 'New Folder' and contains files like 'ass_10', 'ass_11', 'ass_12', etc., and a file named '1_write.c'. The right file manager window also shows the same directory structure. At the bottom, there's a dock with various icons.

```
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; i++){
30         fgets(s, sizeof(s), stdin);
31         fputs(s, file);
32     }
33
34     fclose(file);
35
36     return 0;
37 }
38
```

Solution 2

The screenshot shows a Linux desktop environment with several windows open. In the top left, there's a LibreOffice Calc window titled 'final ass docx - LibreOffice'. Next to it is a terminal window titled 'Linux Lite Terminal' with the command 'cd ~/Desktop/New Folder...'. In the center, there's another terminal window titled 'Linux Lite Terminal' with the command 'g 2_read 2_read.c'. Below these are two file manager windows. The left one is titled 'New Folder' and contains files like 'ass_10', 'ass_11', 'ass_12', etc., and a file named '2_read.c'. The right file manager window also shows the same directory structure. At the bottom, there's a dock with various icons.

```
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 }
```

Solution 3

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

- File Manager:** Shows a directory structure under "/Desktop/Desktop/New Folder/ass_10/file handling". It includes files like "3_count_words.c", "4_copy_file.c", "5_replace_data.c", "test.txt", "4_test.txt", and "4_test2.txt".
- Code Editor:** Sublime Text (UNREGISTERED) showing the content of "3_count_words.c". The code is a C program that reads a file named "test.txt" and prints the number of words and characters. It includes comments explaining the purpose of each section.
- Terminal:** Linux Lite Terminal window titled "Linux Lite Terminal". It shows the command "gcc -o 3_count_words 3_count_words.c" being run, followed by the output of the program which reads "test.txt" and prints "test line 1", "test line 2", "test line 3", and "test line 4". The final output shows the word count (12) and character count (36).

Solution 4

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

- File Manager:** Shows a directory structure under "/Desktop/Desktop/New Folder/ass_10/file handling". It includes files like "3_count_words.c", "4_copy_file.c", "5_replace_data.c", "test.txt", "4_test.txt", and "4_test2.txt".
- Code Editor:** Sublime Text (UNREGISTERED) showing the content of "4_copy_file.c". The code is a C program that copies the content of "test.txt" to "test1.txt". It includes comments and handles errors for source and destination files.
- Terminal:** Linux Lite Terminal window titled "Linux Lite Terminal". It shows the command "gcc -o 4_copy_file 4_copy_file.c" being run, followed by the command "./4_copy_file". It prompts for the source file ("Enter source file name : 4_test.txt") and destination file ("Enter destination file name : 4_test2.txt"). The terminal then displays the contents of "4_test2.txt" which are "test line 1", "test line 2", "test line 3", and "test line 4".

Solution 5

The screenshot shows a Linux desktop environment with a terminal window and a code editor.

Code Editor (Left):

```
#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);
}
```

Terminal (Right):

```
File Edit View Terminal Tabs Help
osboxes ~ New Folder ass_10 file handling gcc -o 5_replace_data 5_replace_data.c
osboxes ~ New Folder ass_10 file handling ./5_replace_data
Input the file name to be opened : test.txt
Input the content for the File : Hello dear
Input the line no you want to replace : 2
osboxes ~ New Folder ass_10 file handling cat test.txt
test line 1
test line 2
Hello dear
osboxes ~ New Folder ass_10 file handling
```