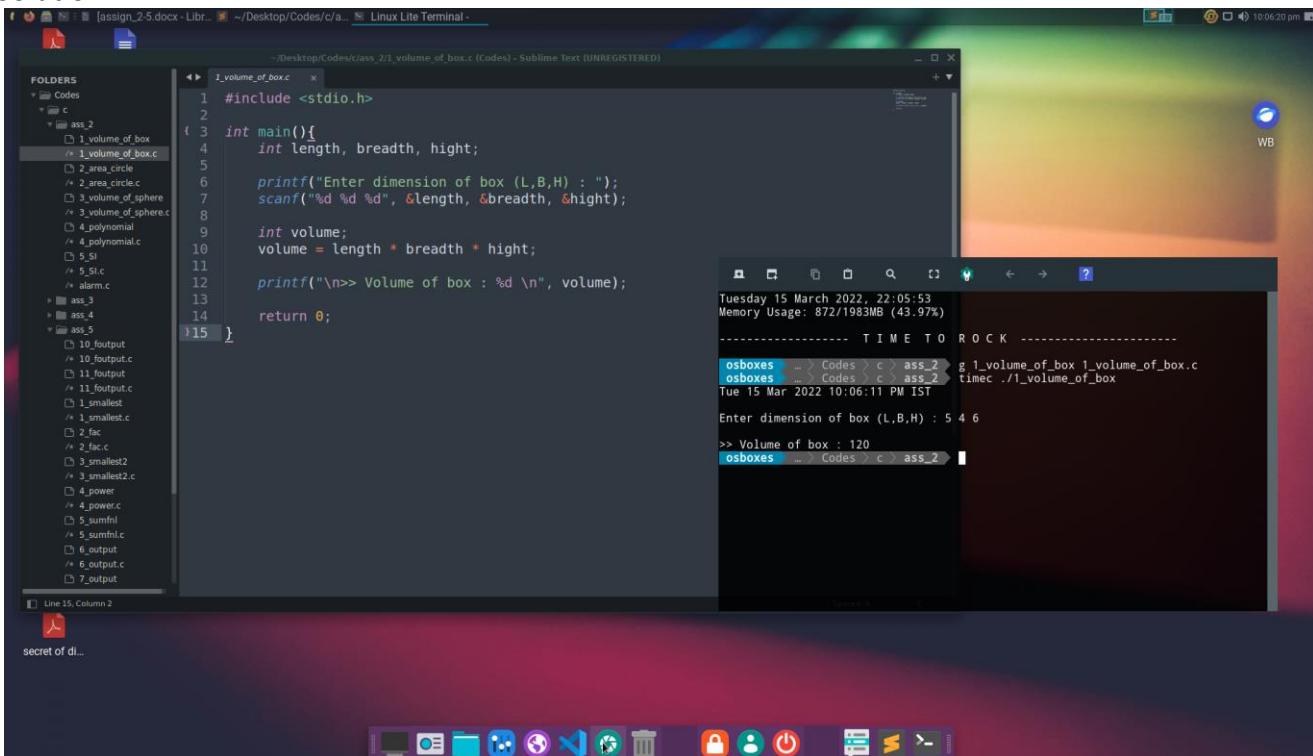


Roll no: 220350320070
Name: MD SAIF
PG-DAC March 2022
Module 01: Basic Programming Concepts

DAY 2

Solution 1:

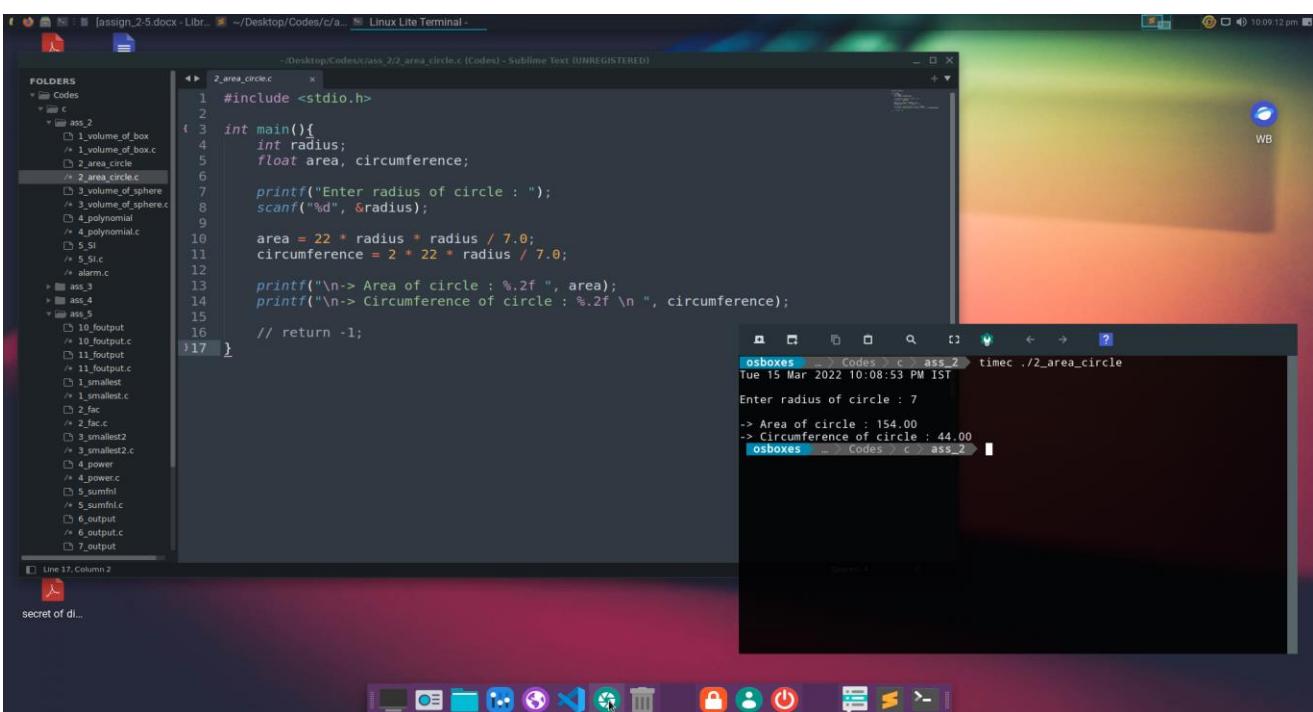


A screenshot of a Linux desktop environment. In the foreground, a terminal window titled "Linux Lite Terminal" is open, showing the command-line interface. In the background, a code editor window titled "Sublime Text (UNREGISTERED)" displays a C program named "1_volume_of_box.c". The code prompts the user for dimensions of a box and calculates its volume. The terminal window shows the execution of the program and its output.

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

Tuesday 15 March 2022, 22:05:53
Memory Usage: 872/1983MB (43.97%)
TIME TO ROCK -----
osboxes ~> Codes > c > ass_2 > g 1_volume_of_box 1_volume_of_box.c
Tue 15 Mar 2022 10:06:11 PM IST
Enter dimension of box (L,B,H) : 5 4 6
-> Volume of box : 120
osboxes ~> Codes > c > ass_2 >
```

Solution 2



A screenshot of a Linux desktop environment. In the foreground, a terminal window titled "Linux Lite Terminal" is open, showing the command-line interface. In the background, a code editor window titled "Sublime Text (UNREGISTERED)" displays a C program named "2_area_circle.c". The code prompts the user for the radius of a circle and calculates its area and circumference. The terminal window shows the execution of the program and its output.

```
#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;
}

osboxes ~> Codes > c > ass_2 > timec ./2_area_circle
Tue 15 Mar 2022 10:08:53 PM IST
Enter radius of circle : 7
-> Area of circle : 154.00
-> Circumference of circle : 44.00
osboxes ~> Codes > c > ass_2 >
```

Solution 3

The screenshot shows a Linux desktop environment with a dark theme. A terminal window titled 'osboxes' is open, showing the command 'timec ./3_volume_of_sphere' and its output: 'Enter radius of sphere : 7' followed by 'Volume of box : 1437.33'. To the left of the terminal is a Sublime Text editor window displaying two files: '2_area_circle.c' and '3_volume_of_sphere.c'. The file '3_volume_of_sphere.c' contains C code for calculating the volume of a sphere. The file tree on the left shows various assignments (ass_1 through ass_5) and their corresponding source files. The bottom of the screen features a dock with various application icons.

```
1 #include <stdio.h>
2 #define PI (float)22/7
3
4 int main(){
5     int radius=10;
6
7     printf("Enter radius of sphere : ");
8     scanf("%d", &radius);
9
10    float volume;
11    volume = (float)4/3 * PI * radius * radius * radius;
12
13    printf("\n>> Volume of box : %.2f \n", volume);
14
15    return 0;
16 }
17 }
```

Solution 4

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

Code Editor:

- File tree:
 - Codes
 - c
 - ass_2
 - 1.volume_of_box
 - 2.area_circle
 - 3.volume_of_sphere
 - 4.polynomial
 - 5_SI.c
 - alarm.c
 - ass_3
 - ass_4
 - ass_5
 - 10_fputut
 - 11_fputut
 - 12_fputut
 - 1.smallest
 - 2.smallest
 - 2.fac
 - 2.fac.c
 - 3.smallest2
 - 3.smallest2.c
 - 4.power
 - 4.power.c
 - 5.summl
 - 5.summlc
 - 6.output
 - 6.output.c

Code content (4_polynomial.c):

```
#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:

```
>> Volume of box : 1437.33
osboxes ... > Codes > c > ass_2 > g 4_polynomial 4_polynomial.c
/usr/bin/ld: /tmp/cc7QLSLM.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+0xa4): 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 > timec ./4_polynomial
Tue 15 Mar 2022 10:16:44 PM IST

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

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

Solution 5

DAY 3

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 Sublime Text window displays a C program named '1_operators.c'. The code performs arithmetic operations on integers and floating-point numbers. To the right, a terminal window titled 'osboxes' shows the output of running the program with inputs 4 and 2, and then with inputs 7 and 3.

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

```
osboxes ~ > Codes > c > ass_2 cd ../ass_3/
osboxes ~ > Codes > c > ass_3 timecc ./1_operators
Tue 15 Mar 2022 10:20:51 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 timecc ./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

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 Sublime Text window displays a C program named '1_operators.c'. The code demonstrates various operators including arithmetic, bitwise, and assignment operators. To the right, a terminal window titled 'osboxes' shows the output of running the program with input 6.

```
#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);
    return 0;
}
```

```
osboxes ~ > Codes > c > ass_3 g 2_bitwise_operators 2_bitwise_operators.c
osboxes ~ > Codes > c > ass_3 timecc ./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 dark theme. A file manager window is open on the left, displaying a directory structure under 'Codes'. In the center, there's a Sublime Text window showing a C program named '3_add_ass.c'. The code prints two numbers and adds them. To the right, a terminal window titled 'Linux Lite Terminal' shows the output of running the program and the compilation of another file.

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

```
osboxes ~ _> Codes <> ass_3 g 2_bitwise_operators 2_bitwise_operators.c
osboxes ~ _> Codes <> 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 <> ass_3 g 3_add_ass 3_add_ass.c
osboxes ~ _> Codes <> 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 <> ass_3
```

Solution 4

The screenshot shows a Linux desktop environment with a dark theme. A file manager window is open on the left, displaying a directory structure under 'Codes'. In the center, there's a Sublime Text window showing a C program named '4_maxOf2.c'. The code compares two numbers and prints the maximum. To the right, a terminal window titled 'Linux Lite Terminal' shows the output of running the program and the compilation of another file.

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

```
osboxes ~ _> Codes <> ass_3 g 2_bitwise_operators 2_bitwise_operators.c
osboxes ~ _> Codes <> 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 <> ass_3 g 3_add_ass 3_add_ass.c
osboxes ~ _> Codes <> 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 <> ass_3 g 4_maxOf2 4_maxOf2.c
osboxes ~ _> Codes <> ass_3 timec ./4_maxOf2
Wed 16 Mar 2022 12:28:21 AM IST
Enter two numbers : 5 3
maximum is 5
osboxes ~ _> Codes <> ass_3
```

Day 4

Solution 1:

A screenshot of a Linux desktop environment. On the left, a Sublime Text window displays a C program named `1_positice_negetive.c`. The code checks if a user input value is positive, negative, or zero. On the right, a terminal window titled "Linux Lite Terminal" shows the output of running the program. It prompts for a value, accepts "8", and prints "8.00 is positive". It then runs the program again with "0" and prints "Number is zero". Finally, it runs the program with "-4" and prints "-4.00 is negative".

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

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

    return 0;
}
```

```
osboxes ~ > Codes > c > ass_4 cd ../ass_4
osboxes ~ > Codes > c > ass_4 g 1_positice_negetive 1_positice_negetive
Wed 16 Mar 2022 12:30:00 AM IST
Enter value : 8
8.00 is positive
osboxes ~ > Codes > c > ass_4 timec ./1_positice_negetive
Wed 16 Mar 2022 12:30:18 AM IST

Enter value : 0
Number is zero
osboxes ~ > Codes > c > ass_4 timec ./1_positice_negetive
Wed 16 Mar 2022 12:30:25 AM IST

Enter value : -4
-4.00 is negative
osboxes ~ > Codes > c > ass_4
```

Solution 2

A screenshot of a Linux desktop environment. On the left, a Sublime Text window displays a C program named `2_greater_num.c`. The program reads three floating-point numbers from the user and prints the largest one. On the right, a terminal window titled "Linux Lite Terminal" shows the output. It prompts for three values, accepts "8", "10", and "12", and prints "12.00 is greatest in all\n". It then runs the program again with "0", "0", and "0" and prints "Number is zero". Finally, it runs the program with "-4", "0", and "0" and prints "-4.00 is negative".

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

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

    return 0;
}
```

```
osboxes ~ > Codes > c > ass_3 cd ../ass_4
osboxes ~ > Codes > c > ass_4 g 1_positice_negetive 1_positice_negetive
Wed 16 Mar 2022 12:30:00 AM IST
Enter value : 8
8.00 is positive
osboxes ~ > Codes > c > ass_4 timec ./1_positice_negetive
Wed 16 Mar 2022 12:30:18 AM IST

Enter value : 0
Number is zero
osboxes ~ > Codes > c > ass_4 timec ./1_positice_negetive
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 Sublime Text window titled "3_smallest_num.c (Codes) - Sublime Text (UNREGISTERED)". The code is as follows:

```
#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;
}
```

The terminal window on the right is titled "Linux Lite Terminal". It shows the command "timec ./3_smallest_num" being run, followed by its output:

```
osboxes ... > Codes > c > ass_4 > g 3_smallest_num 3_smallest_num
Wed 16 Mar 2022 12:35:10 AM IST
Enter 3 numbers : 6 3 9
3 is smallest number
osboxes ... > Codes > c > ass_4 > timec ./3_smallest_num
Wed 16 Mar 2022 12:35:23 AM IST
Enter 3 numbers : 4 8 0
0 is smallest number
osboxes ... > Codes > c > ass_4 >
```

solution 4

A screenshot of a Linux desktop environment. On the left, there is a Sublime Text window titled "4_find_output.c (Codes) - Sublime Text (UNREGISTERED)". The code is as follows:

```
#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 willassing to var i and cond is true bcoz of 1
        printf("CDAC ");
    else
        printf("C Programming ");

    printf("%d\n", i);    // output 5
    return 0;
}
```

The terminal window on the right is titled "Linux Lite Terminal". It shows the command "timec ./4_find_output" being run, followed by its output:

```
osboxes ... > Codes > c > ass_4 > g 3_smallest_num 3_smallest_num
Wed 16 Mar 2022 12:35:10 AM IST
Enter 3 numbers : 6 3 9
3 is smallest number
osboxes ... > Codes > c > ass_4 > timec ./3_smallest_num
Wed 16 Mar 2022 12:35:23 AM IST
Enter 3 numbers : 4 8 0
0 is smallest number
osboxes ... > Codes > c > ass_4 > g 4_find_output 4_find_output.c
osboxes ... > Codes > c > ass_4 > timec ./4_find_output
Wed 16 Mar 2022 12:37:45 AM IST
CDAC 5
osboxes ... > Codes > c > ass_4 >
```

solution 5

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for a weather application showing "12:41 am". Below it is a terminal window titled "Linux Lite Terminal" with the command:

```
osboxes ~ > Codes > c > ass_4 g 5_find_output 5_find_output
osboxes ~ > Codes > c > ass_4 timec ./5_find_output
Wed 16 Mar 2022 12:41:32 AM IST
38
osboxes ~ > Codes > c > ass_4
```

The terminal output shows the result of the program execution:

```
12x18      = 216
216
7 8 9 × ( )
1 2 3 - x² √
0 . % +
```

On the left side of the screen, there is a file manager window titled "Calculator" showing a tree view of files under the "Codes" directory. One file, "5_find_output.c", is open in Sublime Text. The code contains the following C code:

```
#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

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for a weather application showing "12:43 am". Below it is a terminal window titled "Linux Lite Terminal" with the command:

```
osboxes ~ > Codes > c > ass_4 g 6_find_output 6_find_output
osboxes ~ > Codes > c > ass_4 timec ./6_find_output
Wed 16 Mar 2022 12:43:38 AM IST
11
osboxes ~ > Codes > c > ass_4
```

The terminal output shows the result of the program execution:

```
12x18      = 216
216
7 8 9 × ( )
1 2 3 - x² √
0 . % +
```

On the left side of the screen, there is a file manager window titled "Calculator" showing a tree view of files under the "Codes" directory. One file, "6_find_output.c", is open in Sublime Text. The code contains the following C code:

```
#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. In the center is a Sublime Text window displaying a C program named `7_find_output.c`. The code defines a function that calculates the value of $b = 5 * 2 + 2 * 3 \ll h^2 \gg 5:3$ where $h = 9$, resulting in $b = 5$. To the right of the Sublime Text window is a terminal window titled "Linux Lite Terminal" showing the command `g 7_find_output 7_find_output` and its output: `osboxes ... > Codes > c > ass_4 g 7_find_output 7_find_output` and `Wed 16 Mar 2022 12:45:28 AM IST`. Below the terminal is a calculator application showing the calculation $12 \times 18 = 216$.

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

solution 8

A screenshot of a Linux desktop environment. In the center is a Sublime Text window displaying a C program named `8_find_output.c`. The code defines a function that prints "true" if $x=0$ and "false" if $x=1$. To the right of the Sublime Text window is a terminal window titled "Linux Lite Terminal" showing the command `g 8_find_output 8_find_output` and its output: `osboxes ... > Codes > c > ass_4 g 8_find_output 8_find_output` and `Wed 16 Mar 2022 12:47:30 AM IST`. Below the terminal is a calculator application showing the calculation $12 \times 18 = 216$.

```
#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 explorer window titled 'Codes' showing various C source files. In the center, a Sublime Text window displays a C program named '9_find_output.c'. The code contains a main function that prints 'true' if $c > b > a$, and 'false' otherwise. On the right, a terminal window titled 'Linux Lite Terminal' shows the output of running the program: 'false'.

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

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

Solution 10

A screenshot of a Linux desktop environment. On the left, there's a file explorer window titled 'Codes' showing various C source files. In the center, a Sublime Text window displays a C program named '10_find_output.c'. The code contains a main function that initializes variables $a=1$, $b=-1$, $c=0$, and $d=0$. It then increments a and b , and decrements c . It checks the value of d after these operations. If $d=0$, it prints 'false'. If $c<0$ and $d<0$, it prints 'Delhi'. Otherwise, it prints 'Bangalore'. On the right, a terminal window titled 'Linux Lite Terminal' shows the output of running the program: 'Delhi'.

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

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

Day5

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 a C program to find the smallest of three numbers. On the right, a calculator application is visible.

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

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

Solution 2

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 a C program to calculate the factorial of a number. On the right, a calculator application is visible.

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

```
osboxes ... > Codes > c > ass_5 > g 2_fac 2_fac.c
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 >
```

Solution 3

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a directory structure under 'Codes'. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'timec ./3_smallest2' running. On the right, there is a code editor window titled 'assign_2-5.docx - LibreOffice Calc' showing a C program to find the smallest of three numbers.

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

The terminal output shows the program being run with inputs 5, 3, and 8, and the output '3 is the smallest number.'

Solution 4

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a directory structure under 'Codes'. In the center, there is a terminal window titled 'Linux Lite Terminal' with the command 'timec ./4_power' running. On the right, there is a code editor window titled 'assign_2-5.docx - LibreOffice Calc' showing a C program to calculate powers.

```
#include <stdio.h>
int main()
{
    int num, power;
    printf("Enter a base, power : ");
    scanf("%d%d", &num, &power);
    if(num<=0){ printf("\nBase should be greater than 0\n"); return 0; }
    if(power<0){ printf("\nExponent should greater than equal to 0\n"); return 0; }
    int pow=1, temp=power;
    while(power>0){
        pow = pow * num;
    }
    printf("\nResult (%d^%d) : %d\n", num, temp, pow);
    return 0;
}
```

The terminal output shows the program being run with base 5 and power 3, resulting in 125. It also shows error messages for non-positive bases and exponents.

Solution 5

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a directory structure under 'Codes'. In the center, there is a Sublime Text window with multiple tabs open, displaying C code for finding the sum of the first and last digits of a number. On the right, there is a terminal window titled 'Linux Lite Terminal' showing the output of the program. The terminal shows the user entering the number 3746839, and the program outputting the sum of 216.

```
#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;
}
```

```
osboxes ... > Codes > c > ass_5 > g 5_sumfnl 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 >
```

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

Solution 6

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing a directory structure under 'Codes'. In the center, there is a Sublime Text window with multiple tabs open, displaying C code for reversing an integer. On the right, there is a terminal window titled 'Linux Lite Terminal' showing the output of the program. The terminal shows the user entering the integer 3452, and the program outputting the reversed integer 2543.

```
#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;
}
```

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

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

Solution 7

A screenshot of a Linux desktop environment. At the top, there's a window titled "Linux Lite Terminal" showing the command line and output. Below it is a file manager window showing a directory structure under "/Codes". To the right is a calculator application.

Terminal Output:

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

File Manager:

```
Line 10, Column 41
```

Calculator:

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

Solution 8

A screenshot of a Linux desktop environment. At the top, there's a window titled "Linux Lite Terminal" showing the command line and output. Below it is a file manager window showing a directory structure under "/Codes". To the right is a calculator application.

Terminal Output:

```
osboxes ... > Codes > c > ass_5 > g 8_output 8_output.c
osboxes ... > Codes > c > ass_5 > timec ./8_output
Wed 16 Mar 2022 01:39:54 AM IST
1500.00
osboxes ... > Codes > c > ass_5 >
```

File Manager:

```
Line 3, Column 10
```

Calculator:

```
Tab Size: 4
```

Solution 9

The screenshot shows a Linux desktop environment with a dark theme. On the left is a file manager window showing a directory structure under 'Codes'. In the center is a Sublime Text window displaying a C program named '9_output.c'. The code prints the decimal value of a hexadecimal input. To the right is a terminal window titled 'Linux Lite Terminal' showing the output of the program.

```
#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;
}
```

```
osboxes ~ > Codes > c > ass_5 > g 9_output 9_output.c
Wed 16 Mar 2022 01:45:02 AM IST
a=65
osboxes ~ > Codes > c > ass_5 >
```

Solution 10

The screenshot shows a Linux desktop environment with a dark theme. On the left is a file manager window showing a directory structure under 'Codes'. In the center is a Sublime Text window displaying a C program named '10_foutput.c'. The code uses two variables, var1 and var2, to demonstrate a loop. To the right is a terminal window titled 'Linux Lite Terminal' showing the output of the program.

```
#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;
}
```

```
osboxes ~ > Codes > c > ass_5 > g 9_output 9_output.c
osboxes ~ > Codes > c > ass_5 > timec ./9_output
Wed 16 Mar 2022 01:45:02 AM IST
a=65
osboxes ~ > Codes > c > ass_5 > g 10_foutput 10_foutput.c
osboxes ~ > Codes > c > ass_5 > timec ./10_foutput
Wed 16 Mar 2022 01:46:11 AM IST
5 7
osboxes ~ > Codes > c > ass_5 >
```

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 a wireless network. Below it, a window titled "Linux Lite Terminal" is open, showing the command line and its output. To the left of the terminal, a file manager window displays a directory structure under "/Desktop/Codes/c/ass_5". One file in the list is "11_foutput.c", which is currently selected. The code editor window at the bottom also shows "11_foutput.c" with some C code.

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

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

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 a wireless network. Below it, a window titled "Linux Lite Terminal" is open, showing the command line and its output. To the left of the terminal, a file manager window displays a directory structure under "/Desktop/New Folder/ass_5". One file in the list is "12_count_digit.c", which is currently selected. The code editor window at the bottom also shows "12_count_digit.c" with some C code.

```
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 >
```

```
// Write a C program to count total number of digits of an
// Integer number (n).
#include <stdio.h>
int main()
{
    int num;
    printf("Enter the number : ");
    scanf("%d", &num);
    int count=0, temp=num;
    do
    {
        count++;
    } while (num/=10);
    printf("the number %d contains %d digits.\n", temp, count);
    return 0;
}
```

Solution 13

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing various C source files. In the center, there is a terminal window titled "Linux Lite Terminal" with the command line "osboxes ~ > Desktop > New Folder > ass_5 > ./13_powof2". The terminal shows the output of the program, which asks for a number and checks if it is a power of 2. The terminal window has a dark background with light-colored text. The desktop bar at the bottom has icons for a monitor, file manager, terminal, and other system functions.

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

```
osboxes ~ > 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

A screenshot of a Linux desktop environment. On the left, there is a file manager window showing various C source files. In the center, there is a terminal window titled "Linux Lite Terminal" with the command line "osboxes ~ > Desktop > New Folder > ass_5 > ./14_sumofseries". The terminal shows the output of the program, which asks for a number and calculates the sum of the series $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$. The terminal window has a dark background with light-colored text. The desktop bar at the bottom has icons for a monitor, file manager, terminal, and other system functions.

```
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 }
```

```
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 dark theme. In the top bar, there are several application icons and the system tray. A terminal window titled "Linux Lite Terminal" is open, showing the command line and its output. To the left of the terminal, a code editor titled "Sublime Text (UNREGISTERED)" displays a C program named "16_perfect_number.c". The code checks if a given number is a perfect number by summing its proper divisors. Below the code editor, a file browser window shows a directory structure with various files like "ass_6", "ass_7", "ass_8", "ass_9", "j.java", and "pow.c". At the bottom, a dock contains icons for various applications.

```
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 }
```

```
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
```

pdff assignments

Solution 17

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a Java application. The main window is a Sublime Text editor displaying a C program named `17_gcd.c`. The code is a program to find the Greatest Common Divisor (GCD) of two numbers using a 'for' loop. It includes comments explaining the purpose and the algorithm. Below the code, a file browser sidebar shows various assignments and temporary files. In the bottom right corner, there is a terminal window titled "Linux Lite Terminal". The terminal session shows the user running the compiled program `./17_gcd` and entering two numbers at the prompt. The output displays the GCD for different pairs of numbers.

```
17_gcd.c x
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
20     printf("GCD of the numbers %d and %d is %d\n", a, b, gcd);
21     return 0;
22 }
```

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

DAY 6

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including LibreOffice, File Manager, Java, and a terminal icon. The desktop background is a colorful gradient.

Sublime Text Editor: The left pane shows a file tree with a folder named 'ass_6' containing various C files. The right pane displays the code for '1_armstrong.c'. The code is a C program that checks if a given number is Armstrong. It uses a loop to calculate the sum of digits raised to the power of their count and compares it with the original number.

```
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++;
19         num /= 10;
20     }
21
22     //calculate armstrong number from input number
23     num = temp;
24     while(num){
25         arm += pow(num%10, p);
26         num /= 10;
27     }
28
29     //print to tell number is armstrong or not
30     if(temp == arm)
31         printf("%d is an armstrong number.\n", temp);
32     else
33         printf("%d is not an armstrong number.\n", temp);
34
35     return 0;
36 }
```

Terminal Window: The right side of the screen shows a terminal window titled 'Linux Lite Terminal'. It displays the output of running the program. It asks for an input number, then prints whether it is an Armstrong number or not.

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

Solution 2

The screenshot shows a Linux desktop environment with a dark theme, similar to the previous one. In the top panel, there are icons for LibreOffice, File Manager, Java, and a terminal icon. The desktop background is a colorful gradient.

Sublime Text Editor: The left pane shows a file tree with a folder named 'ass_6' containing various C files. The right pane displays the code for '2_palindrome.c'. The code is a C program that checks if a given number is a palindrome by reversing it and comparing 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    return 0;
36 }
```

Terminal Window: The right side of the screen shows a terminal window titled 'Linux Lite Terminal'. It displays the output of running the program. It asks for an input number, then prints whether it is a palindrome or not.

```
osboxes ~ > Desktop > New Folder > ass_6 g_2_palindrome 2_palindrome.c
osboxes ~ > Desktop > New Folder > ass_6 ./2_palindrome
Enter a number :121
121 is palindrome
osboxes ~ > Desktop > New Folder > ass_6 ./2_palindrome
Enter a number :21
21 is not a palindrome
osboxes ~ > Desktop > New Folder > ass_6 ./2_palindrome
Enter a number :11
11 is palindrome
osboxes ~ > Desktop > New Folder > ass_6 ./2_palindrome
Enter a number :4
4 is palindrome
osboxes ~ > Desktop > New Folder > ass_6
```

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 are icons for 'pdf' and 'assignments'. A Sublime Text window is open, showing a file named '3_prime.c'. The code in the file is:

```
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        return 0;
22    }
23 }
```

The terminal window below shows the execution of the program:

```
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 are icons for 'pdf' and 'assignments'. A Sublime Text window is open, showing a file named '4_prime10.c'. The code in the file is:

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

The terminal window below shows the execution of the program:

```
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'. Below the desktop icons, there is a dock with various application icons.

The main window is a Sublime Text editor displaying a C program named `s_find_output.c`. The code finds the maximum value in an array. The terminal window below it shows the command `g 5_find_output s_find_output.c` being run, followed by the output `./5_find_output`.

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

Solution 6

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'. Below the desktop icons, there is a dock with various application icons.

The main window is a Sublime Text editor displaying a C program named `6_find_output.c`. The code prints the values of three variables: `i`, `j`, and `k`. The terminal window below it shows the command `g 6_find_output 6_find_output.c` being run, followed by the output `./6_find_output`.

```
#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 is a 'New Folder' icon. In the bottom left, there are icons for 'pdf' and 'assignments'. A file manager window is open in the background, showing a directory tree with various assignments and files. Two terminal windows are open in the foreground. The left terminal window shows the code for '6_find_output.c' and '7_find_output.c'. The right terminal window shows the command 'g 7_find_output 7_find_output' being run, followed by the output 'IIT MADRAS'. The status bar at the bottom indicates the date and time as 12:33:45 am.

```
6_find_output.c 7_find_output.c
1 #include <stdio.h>
2
3 int main()
4 {
5     int a[6] = {1, 2, 3, 4, 5, 6};
6
7     switch(sizeof(a)){ //return 24, no case matches
8         case 1:
9             case 2:
10            case 3:
11            case 4:
12            case 5: printf("IIT KGP\n");
13            break;
14
15     // print IIT MADRAS
16     printf("IIT MADRAS \n");
17     return 0;
18 }
19

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

Solution 8

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 'WB'. On the desktop, there is a 'New Folder' icon. In the bottom left, there are icons for 'pdf' and 'assignments'. A file manager window is open in the background, showing a directory tree with various assignments and files. Two terminal windows are open in the foreground. The left terminal window shows the code for '8_find_output.c'. The right terminal window shows the command 'g 8_find_output 8_find_output' being run, followed by the output 'a a a a a a'. The status bar at the bottom indicates the date and time as 12:34:31 am.

```
8_find_output.c
1 #include <stdio.h>
2
3 int main()
4 {
5     int i = 0;
6     char c = 'a';
7
8     while(i < 5){
9         i++;
10        switch(c){
11            case 'a': printf("%c\n", c); //5 times
12            break;
13        }
14    }
15    printf("a\n"); //1 time
16
17    // -> total 6 times 'a' will print
18    return 0;
19 }

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

Solution 9

A screenshot of a Linux desktop environment. In the top right corner, there is a system tray icon for 'WB' and a 'New Folder' icon. Below the tray, a terminal window titled 'Linux Lite Terminal' 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
```

To the left of the terminal is a Sublime Text window titled '9_find_output.c'. The code in the editor is:

```
#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, %d\n", a, arr[1]); // a=4, arr[1]=4  
    return 0;  
}
```

The status bar at the bottom of the Sublime Text window indicates 'Line 17, Column 53'. At the bottom of the screen, there is a dock with icons for PDF files and assignments.

Solution 10

A screenshot of a Linux desktop environment. In the top right corner, there is a system tray icon for 'WB' and a 'New Folder' icon. Below the tray, a terminal window titled 'Linux Lite Terminal' shows the command:

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

To the left of the terminal is a Sublime Text window titled '10_lagelement_arr.c'. The code in the editor is:

```
// 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;  
}
```

The status bar at the bottom of the Sublime Text window indicates 'Line 12, Column 21'. At the bottom of the screen, there is a dock with icons for PDF files and assignments.

Solution 11

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 desktop background features a colorful aurora-like pattern. In the center, there is a Sublime Text window titled '11_reverse_arr.c' containing C code to print array elements in reverse order. To the right of the Sublime Text window is a terminal window titled 'Linux Lite Terminal -' showing the command 'g 11_reverse_arr 11_reverse_arr.c' and its output: 'Reverse elements : 6 5 4 3 2 1'. At the bottom of the screen is a dock with various icons.

```
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 }
```

Solution 12

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'. Below it, a 'New Folder' icon is visible. The desktop background features a colorful aurora-like pattern. In the center, there is a Sublime Text window titled '12_merge_arr.c' containing C code to merge two arrays. To the right of the Sublime Text window is a terminal window titled 'Linux Lite Terminal -' showing the command 'g 12_merge_arr 12_merge_arr.c' and its output: '3 15 45 25 3 60 70 80 15 45 25 60 70 80'. At the bottom of the screen is a dock with various icons.

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

Solution 13

A screenshot of a Linux desktop environment. In the top panel, there are several icons including LibreOffice, File Manager, Java, and a terminal icon. The main window is a Sublime Text editor with the file path `/Desktop/New Folder/ass_6/13_dup_del.c`. The code in the editor is:

```
1 // Write a C Program to delete duplicate elements from an array of integers.
2 // First take n as size of first array, then input all n elements of first array.
3 // Now delete duplicate elements from nested array. You can use nested
4 // loop for deleting duplicate elements.
5
6 #include <stdio.h>
7
8 int main()
9 {
10     int t; scanf("%d", &t);
11
12     int a[t];
13     for (int i = 0; i < t; ++i)
14         scanf("%d", &a[i]);
15
16     for (int i = 0; i < t; i++){
17         // if(a[i]==a[t-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                 t--;
23             }
24         }
25         for (int i = 0; i < t; i++)
26             printf("%d ", a[i]);
27     }
28
29     return 0;
30 }
```

The status bar at the bottom shows "Line 22, Column 20". To the right of the editor is a terminal window titled "Linux Lite Terminal" with the command `g 13_dup_del 13_dup_del.c ./13_dup_del` and its output:

```
osboxes ~ Desktop > New Folder > ass_6 g 13_dup_del 13_dup_del.c
6
50
6
7
7
2
2
7
osboxes ~ Desktop > New Folder > ass_6
```

Solution 14

A screenshot of a Linux desktop environment. In the top panel, there are several icons including LibreOffice, File Manager, Java, and a terminal icon. The main window is a Sublime Text editor with the file path `/Desktop/New Folder/ass_6/14_del_elem.c`. The code in the editor is:

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

The status bar at the bottom shows "Line 15, Column 1". To the right of the editor is a terminal window titled "Linux Lite Terminal" with the command `g 14_del_elem 14_del_elem.c ./14_del_elem` and its output:

```
osboxes ~ Desktop > New Folder > ass_6 g 14_del_elem 14_del_elem.c
5
9
10
11
1
20
5
9
10
11
1
osboxes ~ Desktop > New Folder > ass_6
```

Solution 15

The screenshot shows a Linux desktop environment with a green and orange gradient background. In the top right corner, there is a system tray icon for 'WB' and a 'New Folder' icon. On the left side, there is a vertical dock with icons for 'pdf', 'assignments', and other applications.

Sublime Text Editor: The main window title is `./Desktop/New Folder/ass_6/15.sum_ele.c (New Folder) - Sublime Text (UNREGISTERED)`. The code is for a C program to find the sum of all elements of each row of a matrix. It includes comments explaining the logic and an example for a 3x3 matrix. The code uses nested loops to iterate through the matrix and calculate the sum of each row.

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

Terminal Window: The title bar says `Linux Lite Terminal -`. The command `osboxes ~ > Desktop > New Folder > ass_6 g 15_sum_ele 15_sum_ele` is entered, followed by `./15_sum_ele`. The terminal displays the output:

```
4 6
6 7 3
1 2 3
15
16
6
```

Solution 16

The screenshot shows a Linux desktop environment with a green and orange gradient background. In the top right corner, there is a system tray icon for 'WB' and a 'New Folder' icon. On the left side, there is a vertical dock with icons for 'pdf', 'assignments', and other applications.

Sublime Text Editor: The main window title is `./Desktop/New Folder/ass_6/16.subtract.c (New Folder) - Sublime Text (UNREGISTERED)`. The code is for a C program to find subtraction of two matrices. It includes comments explaining the logic and an example for 3x3 matrices. The code uses nested loops to subtract the elements of matrix A from matrix B.

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

Terminal Window: The title bar says `Linux Lite Terminal -`. The command `osboxes ~ > Desktop > New Folder > ass_6 g 16_subtract 16_subtract` is entered, followed by `./16_subtract`. The terminal displays the output:

```
2 3 5
4 5 6
6 5 7
1 5 2
2 3 4
3 3 4
1 -2 3
2 2 2
3 2 3
osboxes ~ > Desktop > New Folder > ass_6
```

Solution 17

The screenshot shows a Linux desktop environment with a dark theme. In the top right corner, there is a system tray icon for 'WB' and a 'New Folder' icon. The desktop background is a colorful abstract image.

At the bottom, there is a dock with various application icons, including a file manager, terminal, and system tools.

A terminal window titled 'Linux Lite Terminal' is open in the bottom right, showing the command line and its 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 7
4 5 6
```

To the left of the terminal, a code editor window titled '17_low_triangle.c' is open. It contains the following C code:

```
1 // Write a C program to print lower triangle of a square matrix.
2 // Input size of square matrix as n. Solve of nxn square matrix
3 // For example the output of a given matrix
4 // 2 3 4 will be 2 0 0
5 // 5 6 7      5 6 0
6 // 4 5 6      4 5 6
7
8 #include <stdio.h>
9
10 int main(){
11     int arr[3][3], k=1;
12
13     for (int i = 0; i < 3; i++){
14         for (int j = 0; j < 3; j++){
15             scanf("%d", &arr[i][j]);
16
17             printf("\n");
18             for (int i = 0; i < 3; i++){
19                 for (int j = 0; j < 3; j++){
20                     if(j<k)
21                         printf("%d ", arr[i][j]);
22                     else
23                         printf("0 ");
24                 }
25             printf("\n"), k++;
26         }
27     }
28     return 0;
}
```

The code editor interface includes a sidebar with file navigation and a status bar at the bottom indicating 'Line 1, Column 1' and 'Spaces: 4'.

DAY 7

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. On the left, a Sublime Text window displays a C program named `l_count_UL.c`. The code counts uppercase and lowercase letters in a given string. On the right, a terminal window titled "Linux Lite Terminal" shows the output of running the program, which correctly prints the uppercase and lowercase counts for the input "Hello from CDAC hyderabad".

```
l_count_UL.c
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     return 0;
}
Line 21, Column 15
parr assignments
```

```
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

The screenshot shows a Linux desktop environment with a dark theme. On the left, a Sublime Text window displays a C program named `2_word_length.c`. The code reads a sentence and finds the largest and smallest words. On the right, a terminal window titled "Linux Lite Terminal" shows the output of running the program, which correctly identifies the largest word as "programming" and the smallest word as "C".

```
2_word_length.c
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         else
25             arr[k][j] = string[i];
26             j++;
27     }
28
29     arr[k][j] = '\0';
30
31     max = strlen(arr[0]);
32     min = strlen(arr[0]);
33
34     for(i = 0 ; i <= k ; i++)
35     {
36         if(max < strlen(arr[i]))
37             max = i;
38
39         if(min > strlen(arr[i]))
40             min = i;
41     }
42
43     printf("\nLargest word is: %s\nSmallest word is: %s\n", arr[max] , arr[min]);
44
45     return 0;
}
Line 25, Column 5
parr assignments
```

```
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. In the top panel, there are several icons and a clock showing 12:59:36 am. Below the panel, there are two windows: a Sublime Text editor and a terminal window.

Sublime Text Editor: The file being edited is `3_reverse.c`. The code is as follows:

```
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
11    printf("Enter the string : ");
12    scanf("%[^\\n]s", d);
13
14    int l = strlen(d)-1;
15
16    printf("The string after reverse : ");
17    while(l >= 0){
18        printf("%c", d[l]);
19        l--;
20    }
21    printf("\n");
22
23    return 0;
24 }
```

Terminal Window: The terminal window is titled "Linux Lite Terminal". It shows the following session:

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

Solution 4

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

Sublime Text Editor: The file being edited is `4_prime_check.c`. The code is as follows:

```
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 }
15
16 int main(){
17     int num;
18     printf("Enter the number : ");
19     scanf("%d", &num);
20
21     if(checkPrime(num))
22         printf("%d is prime number.\n", num);
23     else
24         printf("%d is not a prime number.\n", num);
25
26     return 0;
27 }
```

Terminal Window: The terminal window is titled "Linux Lite Terminal". It shows the following session:

```
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 displays a C program named `5_power_rec.c`. The code is a recursive function to calculate the power of a number. 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 user input "Enter the number and power : 5 3" and the output "5^3: 125".

```
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 displays a C program named `6_binary.c`. The code is a recursive function to print the binary equivalent of a number. On the right, a terminal window titled "Linux Lite Terminal" shows the command `g 6_binary 6_binary.c` being run, followed by the user input "Enter the number : 30" and the output "Binary(30) : 11110".

```
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
16     printf("Enter the number : ");
17     scanf("%d", &num);
18
19     printf("\n Binary(%d) : %d\n", num, binary(num));
20
21     return 0;
22 }
```

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

Solution 7

A screenshot of a Linux Lite desktop environment. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a Java application. The main window is a Sublime Text editor with an open file named `7_find_output.c`. The code contains a `main()` function that declares two character arrays, `str1` and `str2`, and initializes them with specific strings. It then calculates the lengths of these strings and prints them to the terminal. Below the Sublime Text window is a terminal window titled "Linux Lite Terminal". The terminal shows the command `g 7_find_output 7_find_output.c` being run, followed by the output `n1 = 18, n2 = 17`.

```
#include <stdio.h>
int main()
{
    char str1[] = "Week-7-Assignment";
    char str2[] = {"W", 'e', 'e', 'k', '-', '7', '-', 'A', 's', 't', 'r', 'i', 'n', 'g', '\0'};
    int n1 = sizeof(str1)/sizeof(str1[0]); // 18
    int n2 = sizeof(str2)/sizeof(str2[0]); // 17
    printf("n1 = %d, n2 = %d\n", n1, n2); //prints n1 = 18, n2 = 17
    return 0;
}
```

Solution 8

A screenshot of a Linux Lite desktop environment. Similar to the previous solution, it shows a Sublime Text editor with an open file `8_find_output.c` and a terminal window. The code in the Sublime Text editor demonstrates string concatenation. It declares two character arrays, `str1` and `str2`, and initializes `str1` with "hello" and `str2` with " world". It then uses `strcat` to concatenate `str2` onto the end of `str1`, and `strcpy` to copy `str2` into a temporary variable. Finally, it prints the concatenated string "hello world" to the terminal. The terminal window shows the command `g 8_find_output 8_find_output.c` being run, followed by the output `hello world`.

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

Solution 9

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a Java icon. The main window is a Sublime Text editor with an open file named `g_fill_blank.c`. The code is as follows:

```
#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;
}
```

The terminal window below shows the command `g 9_fill_blank g_fill_blank.c` being run, followed by the output `Hwi orea`.

Solution 10

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice icon, a file manager icon, and a Java icon. The main window is a Sublime Text editor with an open file named `10_value_i.c`. The code is as follows:

```
#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;
}
```

The terminal window below shows the command `g 10_value_i 10_value_i.c` being run, followed by the output `0`.

Solution 11

The screenshot shows a Linux desktop environment with a green and orange gradient background. 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 ~ > Desktop > New Folder > ass_7 g 11_find_output 11_find_output". In the center, a Sublime Text window titled "11_find_output.c" is open, displaying C code. The code includes a main function that reverses a character array. The terminal window shows the output "67 67". The bottom of the screen features a dock with various 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
20    // during the reversing elements of array the '\0' null character comes at first
21    // location p[0] = '\0' and '\0' also shows the termination of string
22    // that is the reason the it prints nothing on output console
23    printf("%s", p);
24
25    return 0;
26 }
```

Solution 12

The screenshot shows a Linux desktop environment with a green and orange gradient background. 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 ~ > Desktop > New Folder > ass_7 g 12_find_output 12_find_output". In the center, a Sublime Text window titled "12_find_output.c" is open, displaying C code. The code defines a function "fu" that takes two integers and prints them. The terminal window shows the output "67 67". The bottom of the screen features a dock with various 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
12 void fu(int x, int y){
13     printf("%d, %d\n", x, y); //print 67, 67
14 }
15 }
```

Solution 13

A screenshot of a Linux desktop environment. In the foreground, there is a Sublime Text window titled "-/Desktop/New Folder/ass_7/13_return_value.c (New Folder) - Sublime Text (UNREGISTERED)". The code in the editor is:

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

The terminal window below it is titled "Linux Lite Terminal" and shows the command:

```
osboxes ~ Desktop New Folder ass_7 g 13_return_value 13_return_value
10
osboxes ~ Desktop New Folder ass_7 ./13_return_value
```

Solution 14

A screenshot of a Linux desktop environment. In the foreground, there is a Sublime Text window titled "-/Desktop/New Folder/ass_7/14_count_Hi.c (New Folder) - Sublime Text (UNREGISTERED)". The code in the editor is:

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

The terminal window below it is titled "Linux Lite Terminal" and shows the command:

```
osboxes ~ Desktop New Folder ass_7 g 14_count_Hi 14_count_Hi.c
Hello
osboxes ~ Desktop New Folder ass_7 ./14_count_Hi
```

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' and a 'New Folder' icon. The desktop background is a colorful gradient. In the center, there is a Sublime Text window titled '-/Desktop/New Folder/ass_7/15_find_output.c (New Folder) - Sublime Text (UNREGISTERED)'. The code in the editor is:

```
#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 = 27.08
    return 0;
}
float func(float age[])
{
    int i;
    float result, sum=0.0;
    for (int i = 0; i < 6; ++i){ //count sum of age
        sum += age[i];
    }
    result = (sum/6); //result = avg of age
    return result;
}
```

The status bar at the bottom of the Sublime Text window indicates 'Line 1, Column 1' and 'assignments'. To the right of the Sublime Text window is a terminal window titled 'Linux Lite Terminal -'. The terminal shows the command 'g 15_find_output 15_find_output.c' being run, followed by the output '27.08'.

Solution 16

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 'WB' and a 'New Folder' icon. The desktop background is a colorful gradient. In the center, there is a Sublime Text window titled '-/Desktop/New Folder/ass_7/16_find_output.c (New Folder) - Sublime Text (UNREGISTERED)'. The code in the editor is:

```
#include <stdio.h>
int main()
{
    int i;
    for (int i = 0; i < 5; i++) //loop 5 times
    {
        int i=10;
        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
```

The status bar at the bottom of the Sublime Text window indicates 'Line 15, Column 108' and 'assignments'. To the right of the Sublime Text window is a terminal window titled 'Linux Lite Terminal -'. The terminal shows the command 'g 16_find_output 16_find_output.c' being run, followed by the output '10\n10\n10\n10\n10'.

Solution 17

A screenshot of a Linux desktop environment. On the left, a Sublime Text window displays a C program named `17_count_factexc.c`. The code defines a function `factorial` that calculates the factorial of a number. It includes a check for `n==0` which returns 1. On the right, a terminal window titled "Linux Lite Terminal" shows the command `g 17_count_factexc 17_count_factexc.c` being run, followed by the output `10! = 3628800`.

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

Solution 18

A screenshot of a Linux desktop environment. On the left, a Sublime Text window displays a C program named `18_find_output.c`. The code contains a swap function that swaps two integers. In the main function, it swaps `num1` and `num2` and prints both before and after the swap. The terminal window on the right shows the command `g 18_find_output 18_find_output.c` being run, with the output showing that the values remain 10 and 20 respectively, indicating a pass-by-value call.

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

Solution 19

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

- File Manager:** Shows a tree view of files and folders, including 'ass_5', 'ass_6', 'ass_7', and various C source files like '19_find.c', '10_value_i.c', etc.
- Sublime Text (UNREGISTERED):** An open file named '19_find.c' containing the following C code:

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

- Terminal:** A terminal window titled 'Linux Lite Terminal' showing the command line:

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

DAY 8

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. On the left, there is a file manager window showing a directory structure with files like ass_5, ass_6, ass_7, ass_8, ass_9, j.java, and pow.c. In the center, a Sublime Text window is open with the file 1_multi.c. The code implements multiplication using addition with recursion:

```
// Write a program to implement multiplication using addition. Use recursion.  
#include <stdio.h>  
int multi(int a, int b){ //3  
    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;  
}
```

On the right, a terminal window titled "Linux Lite Terminal" shows the output of running the program:

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

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. On the left, there is a file manager window showing a directory structure with files like ass_5, ass_6, ass_7, ass_8, ass_9, j.java, and pow.c. In the center, a Sublime Text window is open with the file 2_swap.c. The code swaps two numbers:

```
// 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;  
}
```

On the right, a terminal window titled "Linux Lite Terminal" shows the output of running the program:

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

Solution 3

A screenshot of a Linux desktop environment. On the left, a Sublime Text window displays 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 execution of the program. It prompts the user to enter a number (5) and then prints the result (Sum of integers : 15).

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

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

Solution 4

A screenshot of a Linux desktop environment. On the left, a Sublime Text window displays a C program named `4_concat.c`. The code concatenates two strings, "Hello" and " world", into a new string `s`. On the right, a terminal window titled "Linux Lite Terminal" shows the execution of the program. It prints the concatenated string "Hello world".

```
1 // Write a c program to concatenate two strings without using library functions
2
3 #include <stdio.h>
4
5 int main(){
6     char a[] = "Hello", b[] = " world";
7     char s[40];
8
9     for (int i = 0, j = 0, k = 0; j<sizeof(b); ++i)
10    {
11
12        if(a[i] == '\0')
13            continue;
14
15        if(i < sizeof(a))
16            s[k] = a[i], k++;
17
18        else
19            s[k] = b[j], j++, k++;
20
21    }
22
23
24    printf("%s\n", s);
25
26    return 0;
27 }
```

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

Solution 5

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 Sublime Text window is open, showing a file named '5_copy_string.c'. The code in the file is:

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

The terminal window below it shows the command being run and its output:

```
osboxes: ~ > Desktop > New Folder > ass_8 > g 5-copy_string 5_copy_string.c
osboxes: ~ > Desktop > New Folder > ass_8 > ./5_copy_string
Welcome to Future
osboxes: ~ > Desktop > New Folder > ass_8 >
```

Solution 6

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 Sublime Text window is open, showing a file named '6_string_len.c'. The code in the file is:

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

The terminal window below it shows the command being run and its output:

```
osboxes: ~ > Desktop > New Folder > ass_8 > g 6_string_len 6_string_len.c
osboxes: ~ > Desktop > New Folder > ass_8 > ./6_string_len
String : Programmer
size of string : 10
osboxes: ~ > Desktop > New Folder > ass_8 >
```

DAY 9

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a terminal window titled "Linux Lite Terminal". The terminal window displays the command line and the output of a C program named "1_pow.c". The code in "1_pow.c" calculates the power of a number using the `pow` function from the `math.h` library. The terminal shows the compilation of the program and its execution, where it prompts the user for a number and prints the result.

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

```
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+0x5b): undefined reference to `pow'
collect2: error: ld returned 1 exit status
osboxes ~ % Desktop > New Folder > ass_9 g 1_pow 1_pow.c -lm
osboxes ~ % Desktop > New Folder > ass_9 ./1_pow
Enter the number : 2 5
32.000000
osboxes ~ % Desktop > New Folder > ass_9 g 1_pow 1_pow.c -lm
osboxes ~ % Desktop > New Folder > ass_9 ./1_pow
Enter the number : 5 2
25.00
osboxes ~ % Desktop > New Folder > ass_9 ./1_pow
Enter the number : 3 0.5
1.73
osboxes ~ % Desktop > New Folder > ass_9
```

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are several icons including a LibreOffice suite, a file manager, and a terminal window titled "Linux Lite Terminal". The terminal window displays the command line and the output of a C program named "2_leap_yr.c". The code in "2_leap_yr.c" implements a function to determine if a given year is a leap year based on specific rules. The terminal shows the compilation of the program and its execution, where it prompts the user for a year and prints whether it is a leap year or not.

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

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

Solution 3

A screenshot of a Linux desktop environment. In the foreground, there is a Sublime Text window titled "3_fact.c" containing C code for a factorial function. Below it is a terminal window titled "Linux Lite Terminal" showing the execution of the program and its output for different input values.

```
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    return 0;
15 }
16
17 long int factorial(long int num){
18     if(num==1)
19         return 1;
20     else
21         return num * factorial(num-1);
22 }
```

```
osboxes ~ Desktop > New Folder > ass_9 g 3_fact 3_fact
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

A screenshot of a Linux desktop environment. In the foreground, there is a Sublime Text window titled "4_avg.c" containing C code for an average function. Below it is a terminal window titled "Linux Lite Terminal" showing the execution of the program and its output for different input values.

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

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

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 'ass_10' through 'ass_9'. A file named '5_exp.c' is selected.
- Code Editor:** Sublime Text window displaying the content of '5_exp.c'. The code uses the `exp` function from `math.h` to calculate the exponential value of a user-specified number.
- Terminal:** Linux Lite Terminal window showing the compilation and execution of the program. The command `g 5_exp 5_exp.c -lm` is run, followed by the execution command `./5_exp`. The output shows the result of `exp(2.20)`.

Solution 6

The screenshot shows a Linux desktop environment with several windows open. In the foreground, a terminal window titled 'Linux Lite Terminal' is active, displaying the command './6_arr_loc' and its output. The output shows the program's execution and the results of searching for the value 5 in an array. In the background, a code editor window titled 'Sublime Text (UNREGISTERED)' is open, showing a file named '6_arr_loc.c'. The code is a C program that reads an array and counts the occurrences of a specified value. The terminal window also shows the current directory as '/Desktop/New Folder > ass_9'.

```
Line 38, Column 13
Spaces: 4 C

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

Solution 7

The screenshot shows a Linux desktop environment with a dark theme. A Sublime Text window is open, displaying the code for `7_L_search.c`. The code implements a linear search function. A terminal window titled "Linux Lite Terminal" is also open, showing the execution of the program and its output. The terminal shows the array elements 4, 45, 65, 85, 25, 95, and the user's input 95, followed by the message "95 is not present in the array". The terminal window has a blue title bar and a black background.

```
// Write a C program to search a given element from a 1D array and display the position at which it is found by using linear search function. The index location starts from 1.
#include <stdio.h>
#include <stdlib.h>

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

```
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
```

Solution 8

The screenshot shows a Linux desktop environment with a dark theme. A Sublime Text window is open, displaying the code for `8_revse_arr.c`. The code reverses an array using a temporary array. A terminal window titled "Linux Lite Terminal" is also open, showing the execution of the program and its output. The terminal shows the original array elements 8, 9, 10, 6, 4, 7, 11, and the reversed array elements 11, 7, 4, 6, 10, 9, 8. The terminal window has a blue title bar and a black background.

```
// 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]);
    return 0;
}
```

```
osboxes ~ Desktop > New Folder > ass_9 g 8_revse_arr 8_revse_arr.c
osboxes ~ Desktop > New Folder > ass_9 ./8_revse_arr
8
9
10
6
4
7
11
Original array elements are :
8 9 10 6 4 7 11
Reversed array elements are :
11 7 4 6 10 9 8
osboxes ~ Desktop > New Folder > ass_9 ./8_revse_arr
5
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 icons for LibreOffice, File Manager, and a terminal window titled "Linux Lite Terminal". The terminal window shows the command "g 9_swap_arr 9_swap_arr.c" and its output: "Reversed array elements are : 50 60 40 30 20".

The main window is a Sublime Text editor with two tabs open: "8_revse_arr.c" and "9_swap_arr.c". The "9_swap_arr.c" tab contains the following C code:

```
// 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;
}
```

The status bar at the bottom of the Sublime Text window indicates "Line 21, Column 13 - Field 1 of 3" and "Spaces: 4 C".

DAY 10

Structures and Unions

Solution 1

The screenshot shows a Linux desktop environment with a dark theme. On the left, a Sublime Text window displays a C program named `1_std_detail.c`. The code defines a structure `std` with fields `int id`, `char gender`, and `char name[20]`. It includes `#include <stdio.h>` and a `main()` function that prompts for student ID, gender, and name, then prints them. On the right, a terminal window titled "Linux Lite Terminal" shows the command `g 1_std_detail 1_std_detail.c` being run, followed by user input for student ID (101), gender (M), and name (Itachi). The terminal output shows the student details: ID 101, gender M, and name Itachi.

```
// Write a simple structure called student containing fields (int id, char gender and char name[size]); Create a variable of this structure(ex: stdl), fill in the fields of this structure variable and use a pointer to stdl which changes the member values for stdl.
#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;
}
```

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

Solution 2

The screenshot shows a Linux desktop environment with a dark theme. On the left, a Sublime Text window displays a C program named `2_personal.c`. The code defines a structure `personal` with fields `char name[20]`, `char date_of_joining[20]`, and `int salary`. It includes `#include <stdio.h>` and a `main()` function that prompts for 5 people's names, joining dates, and salaries, then prints them. On the right, a terminal window titled "Linux Lite Terminal" shows the command `g 2_personal 2_personal.c` being run. The terminal then prompts for 5 entries: person 1 (name: Naruto, joining: 02-03-1995, salary: 34000), person 2 (name: Ash, joining: 23-11-1998, salary: 31000), person 3 (name: Kirito, joining: 30-05-2001, salary: 33000), person 4 (name: May, joining: 03-06-2002, salary: 23000), and person 5 (name: Asuna, joining: 12-01-2000, salary: 32000). The terminal then prints a summary table for all 5 people.

```
// Define a structure type struct personal that would contain person name, date of joining and salary using this structure to read this information of 5 people and print the same on screen.
#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;
}
```

```
osboxes ~ New Folder > ass_10 struct g 2_personal 2_personal.c
osboxes -> New Folder > ass_10 > struct ./2_personal
osboxes Enter person name 1 : Naruto
Enter date of joining : 02-03-1995
Enter person's salary : 34000

Enter person name 2 : Ash
Enter date of joining : 23-11-1998
Enter person's salary : 31000

Enter person name 3 : Kirito
Enter date of joining : 30-05-2001
Enter person's salary : 33000

Enter person name 4 : May
Enter date of joining : 03-06-2002
Enter person's salary : 23000

Enter person name 5 : Asuna
Enter date of joining : 12-01-2000
Enter person's salary : 32000
-----Details-----
Person name : Naruto
Date of joining : 02-03-1995
Person's salary : 34000

Person name : Ash
Date of joining : 23-11-1998
Person's salary : 31000

Person name : Kirito
Date of joining : 30-05-2001
Person's salary : 33000

Person name : May
Date of joining : 03-06-2002
Person's salary : 23000

Person name : Asuna
Date of joining : 12-01-2000
Person's salary : 32000
osboxes -> New Folder > ass_10 > struct
```

Solution 3

A screenshot of a Linux desktop environment. On the left, a code editor window shows a C program named `3_time.c`. The code defines a `time_struct` with members `hour`, `minute`, and `second`. It includes input/output operations for these values and prints them in the format `16: 40:51`. On the right, a terminal window titled "Linux Lite Terminal" shows the command `g 3_time 3_time.c` being run, followed by user input for hours (21), minutes (44), and seconds (12), and the resulting output: `Time : 21:44:12`.

```
3_time.c
1 // Define structure data type called time_struct containing three member's integer hour, integer
2 // minute and integer second. Develop a program that would assign values to the individual
3 // number and display the time in the following format: 16: 40:51
4
5 #include <stdio.h>
6
7 struct time_struct{
8     int hour, minute, second;
9 };
10
11 int main(){
12     struct time_struct time;
13     printf("Enter Hours : ");
14     scanf("%d", &time.hour);
15     printf("Enter minute : ");
16     scanf("%d", &time.minute);
17     printf("Enter second : ");
18     scanf("%d", &time.second);
19
20     printf("\n Time : %d:%d:%d\n", time.hour, time.minute, time.second);
21
22     return 0;
23 }
```

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

Solution 4

A screenshot of a Linux desktop environment. On the left, a code editor window shows a C program named `4_cricket.c`. The code defines a `cricket` structure with `player_name` and `team_name` fields, and a `batting_avg` float. It reads 50 players and their team-wise details, then prints the team-wise list of players with their batting average. On the right, a terminal window titled "Linux Lite Terminal" shows the command `g 4_cricket 4_cricket.c` being run, followed by user input for player names, team names, and batting averages, and the resulting output: `-----Details-----` followed by the structured data.

```
4_cricket.c
1 // Define a structure called cricket that will describe the following information:
2 // Player name
3 // Team name
4 // Batting average
5
6 // Using cricket, declare an array player with 50 elements and write a C program to read the
7 // information about all the 50 players and print team-wise list containing names of players with
8 // their batting average.
9
10 #include <stdio.h>
11
12 struct cricket{
13     char player_name[20];
14     char team_name[20];
15     float batting_avg;
16 };
17
18 int main(){
19     struct cricket players[50];
20
21     for (int i = 0; i < 3; ++i)
22     {
23         printf("\nEnter player name : ");
24         scanf("%[^n]", players[i].player_name);
25         printf("Enter team name : ");
26         scanf("%[^n]", players[i].team_name);
27         printf("Enter batting avg : ");
28         scanf("%f", &players[i].batting_avg);
29     }
30     printf("\n-----Details-----");
31     for (int i = 0; i < 3; ++i)
32     {
33         printf("\nPlayer name : %s\n", players[i].player_name);
34         printf("Team name : %s\n", players[i].team_name);
35         printf("Batting avg : %.2f\n", players[i].batting_avg);
36     }
37
38     return 0;
39 }
```

```
osboxes ~ -> New Folder > ass_10 > struct > g 4_cricket 4_cricket.c
osboxes 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 >
```

Solution 5

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

Sublime Text Editor: The left pane displays the code for `4_cricket.c` and `5_std_record.c`. The code defines a `student_record` structure and a `main` function to read student data and print it.

```
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 }
```

Terminal Window: The right pane shows the terminal output. It prompts for student names, branch names, and marks, then prints the details for three students.

```
osboxes: ~ > New Folder > ass_10 > struct > g 5_std_record 5_std_record.c
osboxes: ~ > New Folder > ass_10 > struct > ./5_std_record

Enter student name : Rahul
Enter branch name : EE
Enter 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
```

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 "Sublime Text (UNREGISTERED)".

The terminal window displays the following command and output:

```
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
```

The code editor window shows a file named "1_header_var.c" with the following content:

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

Solution 2

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

The terminal window displays the following command and output:

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

The code editor window shows a file named "2_define.c" with the following content:

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

Solution 3

The screenshot shows a Linux desktop environment with a file manager window titled "final ass.docx - LibreOffice Writer" and a terminal window titled "Linux Lite Terminal". The terminal window displays the contents of a C program named "3_loop_macro.c". The code defines a macro "forlo(x, y)" that prints numbers from x to y. The terminal shows the output of the program, which is the numbers 1 through 20 on separate lines.

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

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

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

Solution 4

The screenshot shows a Linux desktop environment with a file manager window titled "final ass.docx - LibreOffice Writer" and a terminal window titled "Linux Lite Terminal". The terminal window displays the contents of a C program named "4_ifdef.c". The code demonstrates conditional compilation using #ifdef, #ifndef, and #endif. It defines two macros, VAR and VAR1, and prints different messages based on their definition status. The terminal shows the output of the program, which includes the definitions of the macros and the resulting printed messages.

```
// Give example usage of conditional compilation using #ifdef, #ifndef and #undef.
#include <stdio.h>
#define VAR
#define VAR1 3

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

    //undefines a preprocessor macro
    #undef VAR

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

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

return 0;

```

```
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 several windows open:

- File Manager:** Shows a folder structure under "/Desktop/New Folder/ass_10/preprocessor directives". Inside are files: 4_cricket.c, 5_std_record.c, 2_define.c, 3_loop_macro.c, and 5_concat.c. The 5_concat.c file is currently selected.
- Terminal:** A "Linux Lite Terminal" window titled "osboxes" is open, showing the command line history:

```
osboxes ... > New Folder > ass_10 > preprocessor directives > g 5_concat 5_concat
at.c
osboxes ... > New Folder > ass_10 > preprocessor directives > ./5_concat
123456
osboxes ... > New Folder > ass_10 > preprocessor directives > 
```
- Code Editor:** A "Sublime Text (UNREGISTERED)" window titled "5_concat.c" is open, displaying the following C code:

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

The status bar at the bottom of the screen indicates "Line 4, Column 28" and "Tab Size: 4".

Storage Class Specifier

Solution 1

A screenshot of a Linux desktop environment. On the left, a file manager window shows a directory structure with files like 'ass_10', 'ass_11', 'ass_12', etc. In the center, a Sublime Text window 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. On the right, a terminal window titled 'Linux Lite Terminal' shows the command 'g 1_static 1_static.c' being run, followed by the output of the program which prints '3'.

```
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

A screenshot of a Linux desktop environment. On the left, a file manager window shows a directory structure with files like 'ass_10', 'ass_11', 'ass_12', etc. In the center, a Sublime Text window displays a C program named '1_static.c'. It includes two other files: '2_EXTERN.c' and '2_EXTERN2.c'. '2_EXTERN.c' contains an 'extern int var;' declaration. '2_EXTERN2.c' contains a definition 'extern int var = 420;'. The 'main()' function in '1_static.c' prints the value of 'var'. On the right, a terminal window titled 'Linux Lite Terminal' shows the command 'g 2_EXTERN.c 2_EXTERN2.c' being run, followed by the output 'var : 420'.

```
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     printf("\nvar : %d\n", var);
13
14     return 0;
15 }
16
17 }
```

Solution 3

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

- File Manager:** Shows a folder structure with files like 1.static.c, 2.extern.c, 3.localvar.c, and 2.extern2.c.
- Code Editor:** Sublime Text (UNREGISTERED) showing the content of 3.localvar.c. The code demonstrates variable scoping with two functions, fun1() and fun2(), each having its own local variable named var.
- Terminal:** Linux Lite Terminal showing the command line path: osboxes ~ > New Folder > ass_10 > storage class specifier > g 3.localvar 3_localvar.c. It also shows the output of a program run with ./3_localvar, which prints "fun1 -> var = 4" and "fun2 -> var = 9".

The desktop bar at the bottom includes icons for PDF viewer, assignments, and system status.

```
1 // Can two functions have variables with the same name?
2 // Give example of them 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
25     fun1();
26     fun2();
27     return 0;
28 }
```

File handling

Solution 1

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window, titled 'Linux Lite Terminal', displays the execution of a C program named '1_write.c'. The user inputs the number of lines to be written (3), and the program outputs 'Line 1', 'Line 2', and 'Line 3' to the terminal. The code editor, titled 'final ass.docx - LibreOffice - ~/Desktop/New Folder...', shows the source code for '1_write.c'. The code is a simple C program that opens 'output.txt' in write mode, reads three lines from standard input, and writes them to the file.

```
// Write a program in C to write multiple lines in a text file.
// Test Data :
// Input the number of lines to be written : 4
// The lines are
// 1
// 2
// 3
// Expected Output :
// The content of the file test.txt is :
// Line 1
// Line 2
// Line 3
// Line 4

#include <stdio.h>
#include <stdlib.h>

int main(){
    FILE *file = fopen("output.txt", "w");
    if (file == NULL){
        printf("fail to open!");
        return -1;
    }
    char s[100]; int line;
    printf("Input the number of lines to be written : ");
    scanf("%d", &line);
    for(int i=0; i<line; i++){
        fgets(s, sizeof(s), stdin);
        fputs(s, file);
    }
    fclose(file);
    return 0;
}
```

Solution 2

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window, titled 'Linux Lite Terminal', displays the execution of a C program named '2_read.c'. The user inputs the file name to be opened ('test.txt'), and the program outputs 'Line 1', 'Line 2', and 'Line 3' to the terminal. The code editor, titled 'final ass.docx - LibreOffice - ~/Desktop/New Folder...', shows the source code for '2_read.c'. The code is a simple C program that opens 'output.txt' in read mode, reads its contents into an array, and then prints the array to the terminal.

```
// Write a program in C to read the file and store the lines into an array.
// Test Data :
// Input the file name to be opened : test.txt
// Expected Output :
// The content of the file output.txt are :
// Line 1
// Line 2
// Line 3

#include <stdio.h>

int main(){
    FILE *file;
    char s[50], c;
    file = fopen("output.txt", "r");
    if(file == NULL){
        printf("File not found\n");
        return -1;
    }
    int i=0;
    while ((c = getc(file)) != EOF){
        s[i] = c; i++;
    }
    printf("%s\n", s);
    return 0;
}
```

Solution 3

The screenshot shows a Linux desktop environment with a dark theme. In the top panel, there are icons for LibreOffice, a file manager, and a terminal window titled "Linux Lite Terminal". The main area contains two windows: Sublime Text and a terminal window.

Sublime Text: The left sidebar shows a file tree with a "New Folder" entry containing several files: ass_10, file handling, 3_count_words.c, 4_copy_file.c, 5_replace_data.c, test.txt, 4_test.txt, and 4_test2.txt. The right pane displays the content of 3_count_words.c. The code is a C program that reads a file named "test.txt" and counts the number of words and characters. It includes comments explaining the purpose of each section and the logic for word and character counting.

Terminal Window: The terminal window has a title bar "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 prints four lines of text from the file "test.txt" and then provides the word and character counts.

Solution 4

The screenshot shows a Linux desktop environment with a dark theme, similar to the previous one. In the top panel, there are icons for LibreOffice, a file manager, and a terminal window titled "Linux Lite Terminal". The main area contains two windows: Sublime Text and a terminal window.

Sublime Text: The left sidebar shows a file tree with a "New Folder" entry containing several files: ass_10, file handling, 3_count_words.c, 4_copy_file.c, 5_replace_data.c, test.txt, 4_test.txt, and 4_test2.txt. The right pane displays the content of 4_copy_file.c. The code is a C program that copies the content of "test.txt" to a new file "test1.txt". It includes comments explaining the purpose of each section and the logic for file reading and writing.

Terminal Window: The terminal window has a title bar "Linux Lite Terminal -". It shows the command "gcc -o 4_copy_file 4_copy_file.c" being run, followed by the command "cat 4_test2.txt" which displays the copied content of "test.txt".

Solution 5

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

- File Manager:** Shows a tree view of files and folders, including 'ass_10' which contains 'file handling' and other sub-folders.
- Code Editor:** Displays the source code for '5_replace_data.c'. The code implements a file replacement program using FILE pointers for reading and writing files.
- Terminal:** Shows the execution of the program. It prompts for the source file ('test.txt'), the content ('Hello dear'), and the line number ('2') to replace. The output shows the original content ('test line 1', 'test line 2'), the replaced content ('Hello dear'), and the final state of the file ('cat test.txt').

```
#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);
}
```

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