

Exp = 1.1

→ Write a C program to print "Hello World"

⇒

```
# include < stdio . h >
mt main ( )
{
    printf (" Hello world ");
    return 0 ;
}
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying a C programming file and its execution output.

The top navigation bar includes File, Edit, Selection, ..., back, forward, search (c programming), and various window control icons.

The left sidebar features icons for file operations like Open, Save, Find, and Settings.

The main workspace shows several tabs for different C files: exp1.3.c, exp2.2.c, exp1.2.c, exp3.3.c, and exp1.1.c (the active tab). The content area contains the following C code:

```
C exp1.1.c > main()
1 #include <stdio.h>
2 int main()
3 {
4     printf("Hello World");
5     return 0;
6 }
```

The right side displays the terminal output:

```
rs\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp1.1.exe'
Hello World
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>
```

The bottom status bar shows file information (Line 6, Col 2), encoding (UTF-8), line endings (CRLF), character count ({} C), and system (Win32).

Exp = 1.2

→ Write a program to print address in new line

→

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

```
int main()
```

{

```
    printf ("Name : Sanchita Luthra \n");
```

```
    printf ("Address : Bidholi \n");
```

```
    printf ("Dehradun, uttarakhand \n");
```

```
    printf ("India \n");
```

```
    return 0
```

}

The screenshot shows a dark-themed code editor interface with several panels:

- Left Sidebar:** Contains icons for File, Edit, Selection, View, Go, and various search and navigation tools.
- Top Bar:** Includes a file browser, a search bar containing "c programming", and standard window control buttons.
- File Explorer:** Shows files like "exp1.3.c", "exp2.2.c", and "exp1.2.c".
- Code Editor:** Displays the content of "exp1.2.c":

```
C exp1.2.c > main()
1 #include <stdio.h>
2 int main()
3 {
4     printf("Name : Sanchita Luthra\n");
5     printf("Address : Bidholi\n");
6     printf("          Dehradun , Uttarakhand\n");
7     printf("          India ");
8     return 0;
9 }
```

- Terminal:** Shows the command-line output of running the program:

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp1.2.exe
Name : Sanchita Luthra
Address : Bidholi
          Dehradun , Uttarakhand
          India
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>
```

- Bottom Bar:** Includes icons for Debug, Compile, Compile & Run, and status information (Ln 9, Col 2, Spaces: 2, UTF-8, CRLF, {}, C, Win32).

$$\text{Exp} = 1.3$$

→ Write a C program that prompts the user to enter their name & age

→

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

```
int main()
```

```
{
```

```
    char name;
```

```
    printf("enter your name \n");
```

```
    scanf("%s", &name);
```

```
    int age;
```

```
    printf("enter your age \n");
```

```
    scanf("%d", &age);
```

```
    return 0;
```

```
}
```

A screenshot of a dark-themed code editor, likely Visual Studio Code, displaying C programming code and its execution output.

The code editor shows three tabs:

- `exp1.3.c`
- `exp2.2.c`
- `exp1.2.c`

The `exp1.3.c` tab contains the following C code:

```
1 #include <stdio.h>
2 int main()
3 {
4     char name;
5     printf("enter your name\n");
6     scanf("%s" , &name);
7
8     int age;
9     printf("enter your age\n");
10    scanf("%d" , &age);
11
12    return 0;
13 }
```

The terminal window shows the execution of the program:

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp1.3.exe
enter your name
Sanchita
enter your age
18
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>
```

At the bottom, the status bar displays:

- Ln 13, Col 2
- Spaces: 4
- UTF-8
- CRLF
- { } C
- Win32

$$\text{Exp} = 1.4$$

→ Write a C program to add two numbers, take number from user

→

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

```
int main()
```

```
{
```

```
int a, b, c;
```

```
printf ("enter first number \n");
```

```
scanf ("%d", &a);
```

```
printf ("enter second number \n");
```

```
scanf ("%d", &b);
```

$$c = a + b$$

```
printf ("the sum of two numbers is : %d \n", c);
```

```
return 0;
```

```
}
```

A screenshot of a dark-themed code editor, likely Visual Studio Code, displaying a C programming file named `exp1.4.c`. The code prompts the user to enter two numbers and prints their sum.

```
C exp1.4.c > main()
1 #include <stdio.h>
2 int main()
3 {
4     int a,b,c;
5
6
7     printf("enter first number\n");
8     scanf("%d" , &a);
9
10    printf("enter second number\n");
11    scanf("%d" , &b);
12
13    c = a + b;
14    printf("the sum of two number is : %d\n" ,c );
15
16    return 0;
17 }
```

The terminal tab shows the execution of the program:

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp1.4.exe
enter first number
26
enter second number
14
the sum of two number is : 40
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>
```

At the bottom, the status bar indicates the current position is Line 17, Column 2, with 2 spaces, using UTF-8 encoding, and the file is a Win32 application.