

LAB FILE
C-PROGRAMMING FILE
GRAPHIC ERA DEEMED TO BE UNIVERSITY



BATCH

2023-2026

BCA (AI & DS)

SUBMITTED BY - AADARSH CHAUDHARY

SUBMITTED TO - MR. RISHI KUMAR

STUDENT ID - 23151260

ROLL NO. - 1

SECTION - I

ASSISTANT PROFESSOR - MS. POOJA CHAHAR

C Program Exercise:

1. WAP for hello world or this is my first C Program.

The screenshot shows a C programming environment with the following details:

- Code Editor:** The file "main.c" contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4
5     printf("AADARSH CHAUDHARY\n");
6
7     printf("Hello, World!\n");
8     return 0;
9 }
10
```
- Output Console:** The output window shows the program's execution:

```
AADARSH CHAUDHARY
Hello, World!

...Program finished with exit code 0
Press ENTER to exit console.
```

2. WAP to add two numbers

The screenshot shows a C programming environment with the following details:

- Code Editor:** The file "main.c" contains the following C code:

```
1 #include<stdio.h>
2
3 int main(){
4     int num1,num2,sum;
5
6     printf("AADARSH CHAUDHARY\n");
7
8     printf("Enter the first number:");
9     scanf("%d",&num1);
10
11    printf("Enter the second number:");
12    scanf("%d",&num2);
13
14    sum = num1 + num2;
15    printf("The sum of two number is: %d\n",sum);
16
17    return 0;
18
19 }
20
```
- Output Console:** The output window shows the program's execution:

```
AADARSH CHAUDHARY
Enter the first number:20
Enter the second number:40
The sum of two number is: 60

...Program finished with exit code 0
Press ENTER to exit console.
```

3. WAP to find area of circle.

The screenshot shows a C IDE interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language setting is set to C. The code editor window contains a file named 'main.c' with the following content:

```
1 #include<stdio.h>
2 int main(){
3     float radius,area;
4     const float pi = 3.14;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter the radius of the circle:");
8     scanf("%f",&radius);
9
10    area = pi* radius * radius;
11
12    printf("Area of the circle is: %.2f\n",area);
13
14    return 0;
15
16 }
17
```

The terminal window below shows the program's output:

```
AADARSH CHAUDHARY
Enter the radius of the circle:50
Area of the circle is: 7850.00

...Program finished with exit code 0
Press ENTER to exit console.
```

4. WAP to divide two numbers.

The screenshot shows a C IDE interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language setting is set to C. The code editor window contains a file named 'main.c' with the following content:

```
1 #include<stdio.h>
2
3 int main(){
4
5     int num1,num2;
6     float result;
7
8     printf("AADARSH CHAUDHARY\n");
9
10    printf("Enter the first num:");
11    scanf("%d",&num1);
12
13    printf("Enter the second number:");
14    scanf("%d",&num2);
15
16    result = num1/num2;
17
18    printf("The result of the division is:%.2f\n",result);
19
20    return 0;
21
22 }
```

The terminal window below shows the program's output:

```
AADARSH CHAUDHARY
Enter the first num:80
Enter the second number:4
The result of the division is:20.00

...Program finished with exit code 0
Press ENTER to exit console.
```

5. WAP to print ASCII value.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor:** File name: main.c. The code prints "AADARSH CHAUDHARY", prompts for a character input, and then prints its ASCII value.

```
1 #include<stdio.h>
2 int main(){
3     char ch;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("enter the character:");
7     scanf("%c",&ch);
8
9     printf("The ASCII value of '%c' is: %d\n",ch,ch);
10
11    return 0;
12 }
13
14 }
```

- Output Window:** Shows the program's execution. It prints "AADARSH CHAUDHARY", asks for a character input ("enter the character:A"), prints the ASCII value ("The ASCII value of 'A' is: 65"), and then exits with code 0.

```
AADARSH CHAUDHARY
enter the character:A
The ASCII value of 'A' is: 65

...Program finished with exit code 0
Press ENTER to exit console.
```

6. WAP to multiply floating point numbers.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor:** File name: main.c. The code prints "AADARSH CHAUDHARY", prompts for two floating-point numbers, multiplies them, and then prints the result.

```
1 #include<stdio.h>
2 int main(){
3     float num1,num2,result;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the first floating-point number:");
7     scanf("%f",&num1);
8
9     printf("Enter the second floating-point number:");
10    scanf("%f",&num2);
11
12    result = num1 * num2;
13    printf("The multiplication result is: %.2f\n",result);
14
15    return 0;
16 }
17
18 }
```

- Output Window:** Shows the program's execution. It prints "AADARSH CHAUDHARY", asks for two floating-point inputs ("Enter the first floating-point number:20" and "Enter the second floating-point number:40"), multiplies them (resulting in 800.00), and then exits with code 0.

```
AADARSH CHAUDHARY
Enter the first floating-point number:20
Enter the second floating-point number:40
The multiplication result is: 800.00

...Program finished with exit code 0
Press ENTER to exit console.
```

7. WAP to SWAP two variables number by using third variable.

The screenshot shows a C IDE interface with a code editor and a terminal window. The code in the editor is:

```
1 #include<stdio.h>
2 int main(){
3     int num1,num2,temp;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the first number:");
7     scanf("%d",&num1);
8
9     printf("Enter the second number:");
10    scanf("%d",&num2);
11
12    printf("Before swapping: num1 = %d,num2 = %d\n",num1,num2);
13
14    //Swapping using third variable
15    temp = num1;
16    num1 = num2;
17    num2 = temp;
18
19    printf("After swapping: num1 = %d,num2 = %d\n",num1,num2);
20    return 0;
21 }
22 }
```

The terminal window below shows the program's output:

```
AADARSH CHAUDHARY
Enter the first number:20
Enter the second number:30
Before swapping: num1 = 20,num2 = 30
After swapping: num1 = 30,num2 = 20

...Program finished with exit code 0
Press ENTER to exit console.
```

8. WAP to SWAP two variables number without using third variable.

The screenshot shows a C IDE interface with a code editor and a terminal window. The code in the editor is:

```
1 #include<stdio.h>
2 int main(){
3     int num1,num2;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the first number:");
7     scanf("%d",&num1);
8
9     printf("Enter the second number:");
10    scanf("%d",&num2);
11
12    printf("Before swapping: num1 = %d,num2 = %d\n",num1,num2);
13    //swapping without using a third variable
14    num1 = num1 + num2;
15    num2 = num1 - num2;
16    num1 = num1 - num2;
17
18    printf("After swapping: num1 = %d,num2 = %d\n",num1,num2);
19
20    return 0;
21 }
22 }
```

The terminal window below shows the program's output:

```
AADARSH CHAUDHARY
Enter the first number:10
Enter the second number:40
Before swapping: num1 = 10,num2 = 40
After swapping: num1 = 40,num2 = 10

...Program finished with exit code 0
Press ENTER to exit console.
```

9. WAP to SWAP three variable numbers without using third variables.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor (main.c):**

```
1 #include<stdio.h>
2 int main
3 {
4     int num1,num2,num3;
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the first number:");
7     scanf("%d",&num1);
8
9     printf("Enter the second number:");
10    scanf("%d",&num2);
11
12    printf("Enter the third number:");
13    scanf("%d",&num3);
14
15    printf("Before swapping: num1 = %d,num2 = %d,num3 = %d\n",num1,num2,num3);
16
17    //swapping without using a third variable
18    num1 = num1 + num2 + num3;
19    num2 = num1 - (num2 + num3);
20    num3 = num1 - (num2 + num3);
21    num1 = num1 - (num2 + num3);
22
23    printf("After swapping: num1 = %d,num2 = %d,num3 = %d\n",num1,num2,num3);
24
25    return 0;
26 }
27 }
```
- Output Window (input):**

```
AADARSH CHAUDHARY
Enter the first number:40
Enter the second number:50
Enter the third number:60
Before swapping: num1 = 40,num2 = 50,num3 = 60
After swapping: num1 = 60,num2 = 40,num3 = 50
```

10. Wap to find the area of rectangle.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor (main.c):**

```
1 #include<stdio.h>
2 int main()
3 {
4     int lenth,width,area;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter the lenth of the rectangle:");
8     scanf("%d",&lenth);
9
10    printf("Enter the width of the rectangle:");
11    scanf("%d",&width);
12
13    area = lenth * width;
14
15    printf("The area of rectangle is: %d\n",area);
16
17    return 0;
18 }
19 
```
- Output Window (input):**

```
AADARSH CHAUDHARY
Enter the lenth of the rectangle:10
Enter the width of the rectangle:20
The area of rectangle is: 200

...Program finished with exit code 0
Press ENTER to exit console.
```

11. WAP to find area of square.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor (main.c):**

```
1 #include<stdio.h>
2 int main(){
3     int side,area;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the lenth of a side of the square:");
7     scanf("%d",&side);
8
9     area = side * side;
10
11    printf("The area of the square is: %d\n",area);
12
13    return 0;
14
15 }
16
```
- Output Window:**

```
AADARSH CHAUDHARY
Enter the lenth of a side of the square:10
The area of the square is: 100

...Program finished with exit code 0
Press ENTER to exit console.
```

12. wap to find area of right angle triangle, isosceles triangle, any triangle with 3 sides.

(I)

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor (main.c):**

```
1 #include<stdio.h>
2 int main(){
3     float base,height;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the base of the right angle triangle:");
7     scanf("%f",&base);
8
9     printf("Enter the height of the right angle triangle:");
10    scanf("%f",&height);
11
12    float area = (base*height)/2;
13    printf("The area of right angle triangle is: %f\n",area);
14
15    return 0;
16
17 }
18
```
- Output Window:**

```
AADARSH CHAUDHARY
Enter the base of the right angle triangle:10
Enter the height of the right angle triangle:20
The area of right angle triangle is: 100.000000

...Program finished with exit code 0
Press ENTER to exit console.
```

(II)

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save), Run, Debug, Stop, Share, Save, Beautify, and Download.
- Code Editor:** The file "main.c" is open, containing the following C code:

```
1 #include<stdio.h>
2 int main(){
3     float base,height;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the base of isosceles triangle:");
7     scanf("%f",&base);
8
9     printf("Enter the height of isosceles triangle:");
10    scanf("%f",&height);
11
12    float area = (base*height)/2;
13
14    printf("The area of isosceles tringle is: %f\n",area);
15
16    return 0;
17
18 }
19
```
- Output Console:** Displays the program's output:

```
AADARSH CHAUDHARY
Enter the base of isosceles triangle:40
Enter the height of isosceles triangle:50
The area of isosceles tringle is: 1000.000000

...Program finished with exit code 0
Press ENTER to exit console.
```

13. wap to find Area and Volume of Cube.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language Selection:** Set to C.
- Code Editor:** File name is "main.c". The code calculates the surface area and volume of a cube based on user input for side length.

```
1 #include<stdio.h>
2 int main(){
3     float side;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the lenth of a side of the cube:");
7     scanf("%f",&side);
8
9     float area = 6 * side * side;
10    float volume = side * side * side;
11    printf("The area of the cube is: %.2f\n",area);
12    printf("The volume of the cube is: %.2f\n",volume);
13
14    return 0;
15
16 }
17
```

- Output Window:** Labeled "input". Displays the program's output:
AADARSH CHAUDHARY
Enter the lenth of a side of the cube:10
The area of the cube is: 600.00
The volume of the cube is: 1000.00

...Program finished with exit code 0
Press ENTER to exit console.

14. wap to find area and volume of cuboid.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language Selection:** Set to C.
- Code Editor:** File name is "main.c". The code calculates the surface area and volume of a cuboid based on user input for length, width, and height.

```
1 #include <stdio.h>
2
3 int main() {
4     float length, width, height;
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the length of the cuboid: ");
7     scanf("%f", &length);
8     printf("Enter the width of the cuboid: ");
9     scanf("%f", &width);
10    printf("Enter the height of the cuboid: ");
11    scanf("%f", &height);
12
13    float area = 2 * (length * width + length * height + width * height);
14    float volume = length * width * height;
15
16    printf("The area of the cuboid is: %.2f\n", area);
17    printf("The volume of the cuboid is: %.2f\n", volume);
18
19    return 0;
20
21 }
```

- Output Window:** Labeled "input". Displays the program's output:
AADARSH CHAUDHARY
Enter the length of the cuboid: 10
Enter the width of the cuboid: 20
Enter the height of the cuboid: 30
The area of the cuboid is: 2200.00
The volume of the cuboid is: 6000.00

...Program finished with exit code 0
Press ENTER to exit console.

15. WAP to find the largest number using the Logical AND operator.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include<stdio.h>
2 int main(){
3     int num1,num2,num3;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter three numbers:");
7     scanf("%d %d %d",&num1,&num2,&num3);
8
9     int largest = (num1 > num2 && num1 > num3) ?num1 : ((num2 > num1 && num2 > num3) ? num2 : num3);
10    printf("The largest number is: %d\n",largest);
11
12    return 0;
13
14 }
15 }
```
- Output Window:** Shows the program's execution:

```
AADARSH CHAUDHARY
Enter three numbers:40 50 20
The largest number is: 50

...Program finished with exit code 0
Press ENTER to exit console.
```

16. WAP to validate the username and password entered by the user is correct or not using the predefined username and password.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5     char username[20];
6     char password[20];
7
8     // Predefined username and password
9     char correctUsername[] = "AADARSH";
10    char correctPassword[] = "CHAUDHARY";
11
12    printf("AADARSH CHAUDHARY\n");
13    printf("Enter username: ");
14    scanf("%s", username);
15
16    printf("Enter password: ");
17    scanf("%s", password);
18
19    if (strcmp(username, correctUsername) == 0 && strcmp(password, correctPassword) == 0) {
20        printf("Username and password are correct. Access granted!\n");
21    } else {
22        printf("Invalid username or password. Access denied!\n");
23    }
24
25    return 0;
26
27 }
```
- Output Window:** Shows the program's execution:

```
AADARSH CHAUDHARY
Enter username: AADARSH
Enter password: CHAUDHARY
Username and password are correct. Access granted!

...Program finished with exit code 0
```

17. WAP to input the positive number from the user to perform the Left shift operator.

The screenshot shows a C IDE interface with a dark theme. The top bar has buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C. The code editor window contains a file named 'main.c' with the following content:

```
1 #include <stdio.h>
2
3 int main() {
4     int num, shift;
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter a positive number: ");
7     scanf("%d", &num);
8
9     printf("Enter the number of positions to shift left: ");
10    scanf("%d", &shift);
11
12    if (num >= 0 && shift >= 0) {
13        int result = num << shift;
14        printf("The result of left shifting %d by %d positions is: %d\n", num, shift, result);
15    } else {
16        printf("Invalid input. Please enter positive numbers only.\n");
17    }
18
19    return 0;
20 }
21
```

The output window below shows the program's execution:

```
AADARSH CHAUDHARY
Enter a positive number: 2
Enter the number of positions to shift left: 4
The result of left shifting 2 by 4 positions is: 32

...Program finished with exit code 0
Press ENTER to exit console.
```

18. WAP to input the positive number from the user to perform the Right shift operator.

The screenshot shows a C IDE interface with a dark theme. The top bar has buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C. The code editor window contains a file named 'main.c' with the following content:

```
1 #include <stdio.h>
2
3 int main() {
4     int number, shift;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a positive number: ");
8     scanf("%d", &number);
9
10    if (number < 0) {
11        printf("Invalid input. Please enter a positive number.\n");
12        return 0;
13    }
14
15    printf("Enter the number of positions to right shift: ");
16    scanf("%d", &shift);
17
18    int result = number >> shift;
19
20    printf("The result of right shifting %d by %d positions is: %d\n", number, shift, result);
21
22    return 0;
23 }
```

The output window below shows the program's execution:

```
AADARSH CHAUDHARY
Enter a positive number: 234
Enter the number of positions to right shift: 2
The result of right shifting 234 by 2 positions is: 58
```

19. WAP to perform the pre increment and pre decrement operator on two integers and print both original value and updated value.

The screenshot shows a C IDE interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to C. The main window displays the file 'main.c' containing the following code:

```
1 #include<stdio.h>
2 int main(){
3     int num1,num2;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the first integer:");
7     scanf("%d",&num1);
8
9     printf("Enter the second integer:");
10    scanf("%d",&num2);
11
12    printf("original values: num1 = %d,num2 = %d\n",num1,num2);
13
14    //pre increment operator
15    ++num1;
16    --num2;
17
18    printf("Updated values: num1 = %d,num2 = %d\n",num1,num2);
19
20    return 0;
21 }
22 }
```

The bottom panel shows the terminal window with the following output:

```
AADARSH CHAUDHARY
Enter the first integer:10
Enter the second integer:20
original values: num1 = 10,num2 = 20
Updated values: num1 = 11,num2 = 19
```

20. WAP to perform the post increment and post decrement operator on two integers and print both original value and updated value.

The screenshot shows a C IDE interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to C. The main window displays the file 'main.c' containing the following code:

```
1 #include <stdio.h>
2 int main() {
3     int num1, num2;
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Enter the first integer: ");
7     scanf("%d",&num1);
8
9     printf("Enter the second integer: ");
10    scanf("%d",&num2);
11
12    printf("Original values: num1 = %d, num2 = %d\n", num1, num2);
13
14    // Post-increment and post-decrement operators
15    num1++; num2--;
16
17    printf("Updated values: num1 = %d, num2 = %d\n", num1, num2);
18    return 0;
19
20 }
21 
```

The bottom panel shows the terminal window with the following output:

```
AADARSH CHAUDHARY
Enter the first integer: 10
Enter the second integer: 20
Original values: num1 = 10, num2 = 20
Updated values: num1 = 11, num2 = 19

...Program finished with exit code 0
Press ENTER to exit console.[]
```

21. WAP for an integer number and to check whether it is divisible by 9 or 7 using OR logical operator.

The screenshot shows a C IDE interface with a code editor and a terminal window. The code editor displays a file named 'main.c' containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int number;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter an integer number: ");
8     scanf("%d", &number);
9
10    if (number % 9 == 0 || number % 7 == 0) {
11        printf("%d is divisible by 9 or 7.\n", number);
12    } else {
13        printf("%d is not divisible by 9 or 7.\n", number);
14    }
15
16    return 0;
17
18 }
19
```

The terminal window below shows the output of running the program. It prompts the user to enter a number, receives '49', and then prints '49 is divisible by 9 or 7.' followed by the standard program finish message.

```
AADARSH CHAUDHARY
Enter an integer number: 49
49 is divisible by 9 or 7.

...Program finished with exit code 0
Press ENTER to exit console.
```

22. WAP to identify gender in single character and print full gender (Ex: if input is 'M' or 'm' – it should print "Male").

The screenshot shows a C IDE interface with a code editor and a terminal window. The code editor displays a file named 'main.c' containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     char gender;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a single character (M/F): ");
8     scanf(" %c", &gender);
9
10    if (gender == 'M' || gender == 'm') {
11        printf("Male\n");
12    } else if (gender == 'F' || gender == 'f') {
13        printf("Female\n");
14    } else {
15        printf("Invalid input\n");
16    }
17
18    return 0;
19 }
20
```

The terminal window below shows the output of running the program. It prompts the user to enter a character, receives 'M', and then prints 'Male' followed by the standard program finish message.

```
AADARSH CHAUDHARY
Enter a single character (M/F): M
Male

...Program finished with exit code 0
Press ENTER to exit console.
```

23. Write a C program to print all natural numbers in reverse (from n to 1).

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify.
- Language:** C.
- Code Editor (main.c):**

```
1 #include <stdio.h>
2
3 int main() {
4     int n;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a positive integer: ");
8     scanf("%d", &n);
9
10    printf("Natural numbers in reverse order: ");
11    for (int i = n; i >= 1; i--) {
12        printf("%d ", i);
13    }
14
15    return 0;
16 }
17
```
- Output Window:**

```
AADARSH CHAUDHARY
Enter a positive integer: 20
Natural numbers in reverse order: 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
...Program finished with exit code 0
Press ENTER to exit console.
```

24. Write a C program to print all alphabets from a to z.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify.
- Language:** C.
- Code Editor (main.c):**

```
1 #include<stdio.h>
2 int main(){
3     char i;
4     printf("AADARSH CHAUDHARY\n");
5     for(i = 'A';i <='Z'; i++){
6         printf("%c\n",i);
7     }
8     return 0;
9 }
```
- Output Window:**

```
AADARSH CHAUDHARY
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
```

25. Write a C program to print all natural numbers from 1 to n.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int n;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a positive integer: ");
8     scanf("%d", &n);
9
10    printf("Natural numbers from 1 to %d: ", n);
11    for (int i = 1; i <= n; i++) {
12        printf("%d ", i);
13    }
14
15    return 0;
16 }
17
```
- Output Console:** Displays the program's output:

```
AADARSH CHAUDHARY
Enter a positive integer: 20
Natural numbers from 1 to 20: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

...Program finished with exit code 0
Press ENTER to exit console.
```

26. program to print all even numbers between 1 to 100.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Even numbers between 1 and 100: ");
7
8     for (int i = 2; i <= 100; i += 2) {
9         printf("%d ", i);
10    }
11
12    return 0;
13 }
14
```
- Output Console:** Displays the program's output:

```
AADARSH CHAUDHARY
Even numbers between 1 and 100: 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100

...Program finished with exit code 0
Press ENTER to exit console.
```

27. Write a C program to print all odd number between 1 to 100.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language Selection:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4
5     printf("AADARSH CHAUDHARY\n");
6     printf("Odd numbers between 1 and 100: ");
7
8     for (int i = 1; i <= 100; i += 2) {
9         printf("%d ", i);
10    }
11
12    return 0;
13 }
```
- Output Console:** Labeled "input" at the top. Displays the output of the program:

```
AADARSH CHAUDHARY
Odd numbers between 1 and 100: 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 5
1 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99

...Program finished with exit code 0
Press ENTER to exit console.
```

28. Write a C program to find sum of all natural numbers between 1 to n.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language Selection:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int n, sum = 0;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a positive integer: ");
8     scanf("%d", &n);
9
10    for (int i = 1; i <= n; i++) {
11        sum += i;
12    }
13
14    printf("The sum of natural numbers from 1 to %d is: %d\n", n, sum);
15
16    return 0;
17 }
18
```
- Output Console:** Labeled "input" at the top. Displays the output of the program:

```
AADARSH CHAUDHARY
Enter a positive integer: 50
The sum of natural numbers from 1 to 50 is: 1275

...Program finished with exit code 0
Press ENTER to exit console.
```

29. Write a C program to find sum of all even numbers between 1 to n.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify.
- Language:** C
- Code Editor (main.c):**

```
1 #include <stdio.h>
2
3 int main() {
4     int n, sum = 0;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a positive integer: ");
8     scanf("%d", &n);
9
10    for (int i = 2; i <= n; i += 2) {
11        sum += i;
12    }
13
14    printf("The sum of even numbers from 1 to %d is: %d\n", n, sum);
15
16    return 0;
17 }
```
- Output Window (input):**

```
AADARSH CHAUDHARY
Enter a positive integer: 20
The sum of even numbers from 1 to 20 is: 110

...Program finished with exit code 0
Press ENTER to exit console.
```

30. Write C program to find sum of all odd numbers between 1 to n.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify.
- Language:** C
- Code Editor (main.c):**

```
1 #include <stdio.h>
2
3 int main() {
4     int n, sum = 0;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a positive integer: ");
8     scanf("%d", &n);
9
10    for (int i = 1; i <= n; i += 2) {
11        sum += i;
12    }
13
14    printf("The sum of odd numbers from 1 to %d is: %d\n", n, sum);
15
16    return 0;
17 }
```
- Output Window (input):**

```
AADARSH CHAUDHARY
Enter a positive integer: 20
The sum of odd numbers from 1 to 20 is: 100

...Program finished with exit code 0
Press ENTER to exit console.
```

31. Write a C program to print multiplication table of any number.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor:** File name: main.c. The code prints "AADARSH CHAUDHARY", prompts for a number, and then prints its multiplication table from 1 to 10.

```
1 #include <stdio.h>
2
3 int main() {
4     int number;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    printf("Multiplication table of %d:\n", number);
11    for (int i = 1; i <= 10; i++) {
12        printf("%d x %d = %d\n", number, i, number * i);
13    }
14
15    return 0;
16 }
```

- Output Console:** Shows the program's execution. It prints "AADARSH CHAUDHARY", asks for a number (10), prints the multiplication table of 10, and then exits.

```
AADARSH CHAUDHARY
Enter a number: 10
Multiplication table of 10:
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

...Program finished with exit code 0
Press ENTER to exit console.
```

32. Write a C program to count number of digits in a number.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor:** File name: main.c. The code prints "AADARSH CHAUDHARY", prompts for a number, and then counts the digits of the entered number.

```
1 #include <stdio.h>
2
3 int main() {
4     int number, count = 0;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    while (number != 0) {
11        number /= 10;
12        count++;
13    }
14
15    printf("The number of digits in the entered number is: %d\n", count);
16
17    return 0;
18 }
```

- Output Console:** Shows the program's execution. It prints "AADARSH CHAUDHARY", asks for a number (987654321), counts the digits, and then exits.

```
AADARSH CHAUDHARY
Enter a number: 987654321
The number of digits in the entered number is: 9

...Program finished with exit code 0
Press ENTER to exit console.
```

33. Write a C program to find first and last digit of a number.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language Selection:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int number, firstDigit, lastDigit;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    lastDigit = number % 10;
11
12    while (number >= 10) {
13        number /= 10;
14    }
15
16    firstDigit = number;
17
18    printf("First digit: %d\n", firstDigit);
19    printf("Last digit: %d\n", lastDigit);
20
21    return 0;
22 }
```
- Output Window:** Labeled "input" at the top. Displays the program's output:

```
AADARSH CHAUDHARY
Enter a number: 123456789
First digit: 1
Last digit: 9

...Program finished with exit code 0
Press ENTER to exit console.
```

34. Write a C program to find sum of first and last digit of a number.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language Selection:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int number, firstDigit, lastDigit, sum;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    lastDigit = number % 10;
11
12    while (number >= 10) {
13        number /= 10;
14    }
15
16    firstDigit = number;
17
18    sum = firstDigit + lastDigit;
19
20    printf("Sum of the first and last digit: %d\n", sum);
21
22    return 0;
23 }
```
- Output Window:** Labeled "input" at the top. Displays the program's output:

```
AADARSH CHAUDHARY
Enter a number: 56789
Sum of the first and last digit: 14

...Program finished with exit code 0
Press ENTER to exit console.
```

35. Write a C program to swap first and last digits of a number.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Language (set to C).
- Code Editor:** File named "main.c" containing C code to swap the first and last digits of a number.
- Output Console:** Displays the program's output:

```
AADARSH CHAUDHARY
Enter a number: 234
Number after swapping first and last digits: 420
```

36. Write a C program to calculate sum of digits of a number.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Language (set to C).
- Code Editor:** File named "main.c" containing C code to calculate the sum of digits of a number.
- Output Console:** Displays the program's output:

```
AADARSH CHAUDHARY
Enter a number: 12345
Sum of the digits: 15

...Program finished with exit code 0
Press ENTER to exit console.
```

37. Write a C program to calculate product of digits of a number.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int number, product = 1, digit;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    while (number > 0) {
11        digit = number % 10;
12        product *= digit;
13        number /= 10;
14    }
15
16    printf("Product of the digits: %d\n", product);
17
18    return 0;
19 }
20
```
- Output Console:** Displays the program's output:

```
AADARSH CHAUDHARY
Enter a number: 45
Product of the digits: 20

...Program finished with exit code 0
Press ENTER to exit console.
```

38. Write a C program to enter a number and print its reverse.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow.
- Language:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int number, reverse = 0, remainder;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    while (number != 0) {
11        remainder = number % 10;
12        reverse = reverse * 10 + remainder;
13        number /= 10;
14    }
15
16    printf("The reverse of the number is: %d\n", reverse);
17
18    return 0;
19 }
20
```
- Output Console:** Displays the program's output:

```
AADARSH CHAUDHARY
Enter a number: 56
The reverse of the number is: 65

...Program finished with exit code 0
Press ENTER to exit console.
```

39. Write a C program to check whether a number is palindrome or not.

The screenshot shows a C IDE interface with a code editor and a terminal window. The code editor displays a C program named 'main.c' that checks if a given number is a palindrome. The terminal window shows the output of running the program with the input '232', which correctly identifies it as a palindrome.

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int number, originalNumber, reverse = 0, remainder;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    originalNumber = number;
11
12    while (number != 0) {
13        remainder = number % 10;
14        reverse = reverse * 10 + remainder;
15        number /= 10;
16    }
17
18    if (originalNumber == reverse) {
19        printf("%d is a palindrome number.\n", originalNumber);
20    } else {
21        printf("%d is not a palindrome number.\n", originalNumber);
22    }
23
24    return 0;
25 }
```

input
Enter a number: 232
232 is a palindrome number.

40. Write a C program to find frequency of each digit in a given integer.

The screenshot shows a C IDE interface with a code editor and a terminal window. The code editor displays a C program named 'main.c' that calculates the frequency of each digit in a given integer. The terminal window shows the output of running the program with the input '122333', displaying the frequency of each digit (1: 1, 2: 2, 3: 3).

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int number, digit, frequency[10] = {0};
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    while (number != 0) {
11        digit = number % 10;
12        frequency[digit]++;
13        number /= 10;
14    }
15
16    printf("Frequency of each digit:\n");
17
18    for (int i = 0; i < 10; i++) {
19        if (frequency[i] > 0) {
20            printf("Digit %d: %d\n", i, frequency[i]);
21        }
22    }
23
24    return 0;
25 }
```

input
AADARSH CHAUDHARY
Enter a number: 122333
Frequency of each digit:
Digit 1: 1
Digit 2: 2
Digit 3: 3

41. Write a C program to enter a number and print it in words.

The screenshot shows a C IDE interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The status bar at the top right says "Language C". The code editor window is titled "main.c" and contains the following C code:

```
1 #include <stdio.h>
2
3 void printNumberInWords(int number) {
4     if (number == 0) {
5         printf("Zero");
6         return;
7     }
8
9     const char *ones[] = {"", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};
10    const char *tens[] = {"", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};
11    const char *teens[] = {"Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};
12
13    if (number >= 100) {
14        printf("%s Hundred ", ones[number / 100]);
15        number %= 100;
16    }
17
18    if (number >= 20) {
19        printf("%s ", tens[number / 10]);
20        number %= 10;
21    } else if (number >= 10) {
22        printf("%s", teens[number - 10]);
23        return;
24    }
25
26    if (number > 0) {
27        printf("%s", ones[number]);
28    }
29 }
30
31 int main() {
32     int number;
33
34     printf("AADARSH CHAUDHARY\n");
35     printf("Enter a number: ");
36     scanf("%d", &number);
37
38     printf("Number in words: ");
39     printNumberInWords(number);
40     printf("\n");
41
42     return 0;
43 }
```

The terminal window below the code editor shows the output of the program:

```
AADARSH CHAUDHARY
Enter a number: 123
Number in words: One Hundred Twenty Three
```

42. Write a C program to print all ASCII character with their values.

The screenshot shows a C IDE interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The status bar at the top right says "Language C". The code editor window is titled "main.c" and contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int i;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("ASCII Characters with their Values:\n");
8
9     for (i = 0; i <= 127; i++) {
10         printf("Character: %c\t ASCII Value: %d\n", i, i);
11     }
12
13     return 0;
14 }
```

AADARSH CHAUDHARY

ASCII Characters with their Values:

```
Character:      ASCII Value: 0
Character:      ASCII Value: 1
Character:      ASCII Value: 2
Character:      ASCII Value: 3
Character:      ASCII Value: 4
Character:      ASCII Value: 5
Character:      ASCII Value: 6
Character:      ASCII Value: 7
Character:      ASCII Value: 8
Character:          ASCII Value: 9
Character:
Character:      ASCII Value: 10
Character:
Character:      ASCII Value: 11
Character:
Character:      ASCII Value: 12
Character:      ASCII Value: 13
Character:      ASCII Value: 14
Character:      ASCII Value: 15
Character:      ASCII Value: 16
Character:      ASCII Value: 17
Character:      ASCII Value: 18
Character:      ASCII Value: 19
Character:      ASCII Value: 20
Character:      ASCII Value: 21
Character:      ASCII Value: 22
Character:      ASCII Value: 23
Character:      ASCII Value: 24
Character:      ASCII Value: 25
Character:      ASCII Value: 26
Character:      ASCII Value: 27
Character:      ASCII Value: 28
Character:      ASCII Value: 29
Character:      ASCII Value: 30
Character:      ASCII Value: 31
Character:      ASCII Value: 32
Character: !    ASCII Value: 33
Character: "    ASCII Value: 34
Character: #    ASCII Value: 35
Character: $    ASCII Value: 36
Character: %    ASCII Value: 37
Character: &   ASCII Value: 38
Character: '    ASCII Value: 39
Character: (    ASCII Value: 40
Character: )    ASCII Value: 41
Character: *    ASCII Value: 42
Character: +    ASCII Value: 43
```

```
Character: ,      ASCII Value: 44
Character: -      ASCII Value: 45
Character: .      ASCII Value: 46
Character: /      ASCII Value: 47
Character: 0      ASCII Value: 48
Character: 1      ASCII Value: 49
Character: 2      ASCII Value: 50
Character: 3      ASCII Value: 51
Character: 4      ASCII Value: 52
Character: 5      ASCII Value: 53
Character: 6      ASCII Value: 54
Character: 7      ASCII Value: 55
Character: 8      ASCII Value: 56
Character: 9      ASCII Value: 57
Character: :      ASCII Value: 58
Character: ;      ASCII Value: 59
Character: <      ASCII Value: 60
Character: =      ASCII Value: 61
Character: >      ASCII Value: 62
Character: ?      ASCII Value: 63
Character: @      ASCII Value: 64
Character: A      ASCII Value: 65
Character: B      ASCII Value: 66
Character: C      ASCII Value: 67
Character: D      ASCII Value: 68
Character: E      ASCII Value: 69
Character: F      ASCII Value: 70
Character: G      ASCII Value: 71
Character: H      ASCII Value: 72
Character: I      ASCII Value: 73
Character: J      ASCII Value: 74
Character: K      ASCII Value: 75
Character: L      ASCII Value: 76
Character: M      ASCII Value: 77
Character: N      ASCII Value: 78
Character: O      ASCII Value: 79
Character: P      ASCII Value: 80
Character: Q      ASCII Value: 81
Character: R      ASCII Value: 82
Character: S      ASCII Value: 83
Character: T      ASCII Value: 84
Character: U      ASCII Value: 85
Character: V      ASCII Value: 86
Character: W      ASCII Value: 87
Character: X      ASCII Value: 88
Character: Y      ASCII Value: 89
Character: Z      ASCII Value: 90
Character: [      ASCII Value: 91
Character: \      ASCII Value: 92
```

```
Character: ]      ASCII Value: 93
Character: ^      ASCII Value: 94
Character: _      ASCII Value: 95
Character: `      ASCII Value: 96
Character: a      ASCII Value: 97
Character: b      ASCII Value: 98
Character: c      ASCII Value: 99
Character: d      ASCII Value: 100
Character: e      ASCII Value: 101
Character: f      ASCII Value: 102
Character: g      ASCII Value: 103
Character: h      ASCII Value: 104
Character: i      ASCII Value: 105
Character: j      ASCII Value: 106
Character: k      ASCII Value: 107
Character: l      ASCII Value: 108
Character: m      ASCII Value: 109
Character: n      ASCII Value: 110
Character: o      ASCII Value: 111
Character: p      ASCII Value: 112
Character: q      ASCII Value: 113
Character: r      ASCII Value: 114
Character: s      ASCII Value: 115
Character: t      ASCII Value: 116
Character: u      ASCII Value: 117
Character: v      ASCII Value: 118
Character: w      ASCII Value: 119
Character: x      ASCII Value: 120
Character: y      ASCII Value: 121
Character: z      ASCII Value: 122
Character: {      ASCII Value: 123
Character: |      ASCII Value: 124
Character: }      ASCII Value: 125
Character: ~      ASCII Value: 126
Character:       ASCII Value: 127
```

```
...Program finished with exit code 0
Press ENTER to exit console.[]
```

43. Write a C program to find power of a number using for loop.

The screenshot shows a C IDE interface with the following details:

- Title Bar:** Includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon.
- Language Selection:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int power(int base, int exponent) {
4     int result = 1;
5
6     for (int i = 0; i < exponent; i++) {
7         result *= base;
8     }
9
10    return result;
11 }
12
13 int main() {
14     int base, exponent;
15
16     printf("AADARSH CHAUDHARY\n");
17     printf("Enter the base number: ");
18     scanf("%d", &base);
19
20     printf("Enter the exponent: ");
21     scanf("%d", &exponent);
22
23     int result = power(base, exponent);
24
25     printf("%d raised to the power of %d is %d\n", base, exponent, result);
26
27     return 0;
28 }
```
- Output Console:** Labeled "input" at the top. Displays the program's output:

```
AADARSH CHAUDHARY
Enter the base number: 4
Enter the exponent: 5
4 raised to the power of 5 is 1024
```

44. Write a C program to find all factors of a number.

The screenshot shows a C IDE interface with the following details:

- Title Bar:** Includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon.
- Language Selection:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int number;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter a number: ");
8     scanf("%d", &number);
9
10    printf("Factors of %d are: ", number);
11
12    for (int i = 1; i <= number; i++) {
13        if (number % i == 0) {
14            printf("%d ", i);
15        }
16    }
17
18    printf("\n");
19
20    return 0;
21 }
22
```
- Output Console:** Labeled "input" at the top. Displays the program's output:

```
AADARSH CHAUDHARY
Enter a number: 20
Factors of 20 are: 1 2 4 5 10 20

...Program finished with exit code 0
Press ENTER to exit console.
```

45. Write a C program to calculate factorial of a number.

The screenshot shows a C IDE interface with a code editor and a terminal window. The code editor displays a C program named 'main.c' that calculates the factorial of a given number. The terminal window shows the program's execution, including user input and the resulting output.

```
1 #include <stdio.h>
2
3 int factorial(int number) {
4     int result = 1;
5
6     for (int i = 1; i <= number; i++) {
7         result *= i;
8     }
9
10    return result;
11 }
12
13 int main() {
14     int number;
15
16     printf("AADARSH CHAUDHARY\nEnter a number: ");
17     scanf("%d", &number);
18
19     int result = factorial(number);
20
21     printf("The factorial of %d is %d\n", number, result);
22
23     return 0;
24 }
```

AADARSH CHAUDHARY
Enter a number: 25
The factorial of 25 is 2076180480

...Program finished with exit code 0
Press ENTER to exit console.

46. Write a C program to find HCF (GCD) of two numbers.

The screenshot shows a C IDE interface with a code editor and a terminal window. The code editor displays a C program named 'main.c' that finds the HCF (GCD) of two numbers. The terminal window shows the program's execution, including user input and the resulting output.

```
1 #include <stdio.h>
2
3 int findHCF(int num1, int num2) {
4     int hcf;
5
6     for (int i = 1; i <= num1 && i <= num2; i++) {
7         if (num1 % i == 0 && num2 % i == 0) {
8             hcf = i;
9         }
10    }
11
12    return hcf;
13 }
14
15 int main() {
16     int num1, num2;
17
18     printf("AADARSH CHAUDHARY\nEnter the first number: ");
19     scanf("%d", &num1);
20
21     printf("Enter the second number: ");
22     scanf("%d", &num2);
23
24     int hcf = findHCF(num1, num2);
25
26     printf("The HCF of %d and %d is %d\n", num1, num2, hcf);
27
28     return 0;
29 }
```

AADARSH CHAUDHARY
Enter the first number: 6
Enter the second number: 8
The HCF of 6 and 8 is 2

47. Write a C program to find LCM of two numbers.

The screenshot shows a C IDE interface with the following details:

- Title Bar:** Includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon.
- Language:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int findLCM(int num1, int num2) {
4     int max;
5
6     max = (num1 > num2) ? num1 : num2;
7
8     while (1) {
9         if (max % num1 == 0 && max % num2 == 0) {
10            return max;
11        }
12        max++;
13    }
14 }
15
16 int main() {
17     int num1, num2;
18
19     printf("AADARSH CHAUDHARY\nEnter the first number: ");
20     scanf("%d", &num1);
21
22     printf("Enter the second number: ");
23     scanf("%d", &num2);
24
25     int lcm = findLCM(num1, num2);
26
27     printf("The LCM of %d and %d is %d\n", num1, num2, lcm);
28
29     return 0;
30 }
```
- Output Window:** Shows the program's execution:

```
AADARSH CHAUDHARY
Enter the first number: 8
Enter the second number: 10
The LCM of 8 and 10 is 40
```

48. Write a C program to check whether a number is Prime number or not.

The screenshot shows a C IDE interface with the following details:

- Title Bar:** Includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon.
- Language:** Set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int checkPrime(int num) {
4     int i;
5
6     if (num <= 1) {
7         return 0;
8     }
9
10    for (i = 2; i * i <= num; i++) {
11        if (num % i == 0) {
12            return 0;
13        }
14    }
15
16    return 1;
17 }
18
19 int main() {
20     int num;
21
22     printf("AADARSH CHAUDHARY\nEnter a number: ");
23     scanf("%d", &num);
24
25     if (checkPrime(num)) {
26         printf("%d is a prime number!\n", num);
27     } else {
28         printf("%d is not a prime number.\n", num);
29     }
30
31     return 0;
32 }
```
- Output Window:** Shows the program's execution:

```
AADARSH CHAUDHARY
Enter a number: 17
17 is a prime number!
```

49. Write a C program to print all Prime numbers between 1 to n.

The screenshot shows a C IDE interface with the following details:

- Title Bar:** Includes standard icons for Run, Debug, Stop, Share, Save, and Beautify, along with a Language dropdown set to C.
- Code Editor:** The file is named "main.c". The code defines a function `isPrime` that checks if a number is prime by testing divisibility from 2 to \sqrt{n} . It then calls `printPrimeNumbers` for the range [1, n].
- Output Window:** Labeled "input", it displays the program's execution:
 - User input: AADARSH CHAUDHARY
 - User input: Enter a number: 20
 - Program output: Prime numbers between 1 and 20 are:
 - Program output: 2 3 5 7 11 13 17 19

50. Write a C program to find sum of all prime numbers between 1 to n.

The screenshot shows a C IDE interface with the following details:

- Title Bar:** Includes standard icons for Run, Debug, Stop, Share, Save, and Beautify, along with a Language dropdown set to C.
- Code Editor:** The file is named "main.c". The code defines a function `sumOfPrimes` that iterates through numbers from 2 to n, adding each prime number to a running total. It then calls `sumOfPrimes` and prints the result.
- Output Window:** Labeled "input", it displays the program's execution:
 - User input: AADARSH CHAUDHARY
 - User input: Enter a number: 100
 - Program output: The sum of all prime numbers between 1 and 100 is: 1060

51. Write a C program to find all prime factors of a number.

The screenshot shows a code editor interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language setting is set to C. The code file is named 'main.c'. The code defines a function 'primeFactors' that prints the prime factors of a given number. It then calls this function from the 'main' function, which reads a number from the user and prints its prime factors. The terminal window below shows the execution of the program, with the user input '10' and the output 'Prime factors of 10 are: 2 5'.

```
1 #include <stdio.h>
2
3 void primeFactors(int num) {
4     printf("Prime factors of %d are: ", num);
5
6     for (int i = 2; i <= num; i++) {
7         while (num % i == 0) {
8             printf("%d ", i);
9             num /= i;
10        }
11    }
12
13    printf("\n");
14 }
15
16 int main() {
17     int number;
18     printf("AADARSH CHAUDHARY\nEnter a number: ");
19     scanf("%d", &number);
20
21     primeFactors(number);
22
23     return 0;
24 }
25
```

AADARSH CHAUDHARY
Enter a number: 10
Prime factors of 10 are: 2 5

...Program finished with exit code 0
Press ENTER to exit console.

52. Write a C program to check whether a number is Armstrong number or not.

The screenshot shows a code editor interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language setting is set to C. The code file is named 'main.c'. It contains a function 'isArmstrong' that calculates the sum of the digits of a number raised to the power of the number of digits. It then compares this sum with the original number to determine if it is an Armstrong number. The 'main' function uses this function to check if the user input '153' is an Armstrong number, printing the result to the terminal. The terminal window shows the user input '153' and the output '153 is an Armstrong number.'

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int isArmstrong(int num) {
5     int originalNum, remainder, result = 0, n = 0;
6
7     originalNum = num;
8
9     while (originalNum != 0) {
10        originalNum /= 10;
11        ++n;
12    }
13
14    originalNum = num;
15
16    while (originalNum != 0) {
17        remainder = originalNum % 10;
18        result += pow(remainder, n);
19        originalNum /= 10;
20    }
21
22    if (result == num)
23        return 1;
24    else
25        return 0;
26 }
27
28 int main() {
29     int number;
30     printf("AADARSH CHAUDHARY\nEnter a number: ");
31     scanf("%d", &number);
32
33     if (isArmstrong(number))
34         printf("%d is an Armstrong number.\n", number);
35     else
36         printf("%d is not an Armstrong number.\n", number);
37
38     return 0;
39 }
```

AADARSH CHAUDHARY
Enter a number: 153
153 is an Armstrong number.

53. Write a C program to print all Armstrong numbers between 1 to n.

```
main.c
1 #include <stdio.h>
2 #include <math.h>
3
4 int isArmstrong(int num) {
5     int originalNum, remainder, result = 0, n = 0;
6
7     originalNum = num;
8
9     while (originalNum != 0) {
10        originalNum /= 10;
11        ++n;
12    }
13
14    originalNum = num;
15
16    while (originalNum != 0) {
17        remainder = originalNum % 10;
18        result += pow(remainder, n);
19        originalNum /= 10;
20    }
21
22    if (result == num)
23        return 1;
24    else
25        return 0;
26 }
27
28 void printArmstrongNumbers(int n) {
29     printf("Armstrong numbers between 1 and %d are:\n", n);
30
31     for (int i = 1; i <= n; i++) {
32         if (isArmstrong(i))
33             printf("%d ", i);
34     }
35
36     printf("\n");
37 }
38
39 int main() {
40     int number;
41     printf("AADARSH CHAUDHARY\nEnter a number: ");
42     scanf("%d", &number);
43
44     printArmstrongNumbers(number);
45
46     return 0;
47 }
```

```
AADARSH CHAUDHARY
Enter a number: 100
Armstrong numbers between 1 and 100 are:
1 2 3 4 5 6 7 8 9

...Program finished with exit code 0
Press ENTER to exit console.
```

54. Write a C program to check whether a number is Perfect number or not.

```
main.c
1 #include <stdio.h>
2
3 int isPerfectNumber(int num) {
4     int sum = 0;
5
6     for (int i = 1; i <= num / 2; i++) {
7         if (num % i == 0) {
8             sum += i;
9         }
10    }
11
12    if (sum == num)
13        return 1;
14    else
15        return 0;
16 }
17
18 int main() {
19     int number;
20     printf("AADARSH CHAUDHARY\nEnter a number: ");
21     scanf("%d", &number);
22
23     if (isPerfectNumber(number))
24         printf("%d is a Perfect number.\n", number);
25     else
26         printf("%d is not a Perfect number.\n", number);
27
28     return 0;
29 }
```

```
AADARSH CHAUDHARY
Enter a number: 6
6 is a Perfect number.
```

55. Write a C program to print all Perfect numbers between 1 to n.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify.
- Code Editor:** File named "main.c". The code defines a function `isPerfectNumber` that checks if a number is perfect by summing its divisors. It also contains a `printPerfectNumbers` function that prints all perfect numbers up to a given limit. The `main` function prompts the user for a number and calls the printing function.
- Output Window:** Labeled "input". It displays the user input "AADARSH CHAUDHARY" and the program's output: "Enter a number: 100", "Perfect numbers between 1 and 100 are:", and "6 28".

```
1 #include <stdio.h>
2
3 int isPerfectNumber(int num) {
4     int sum = 0;
5     for (int i = 1; i < num; i++) {
6         if (num % i == 0) {
7             sum += i;
8         }
9     }
10    if (sum == num) {
11        return 1;
12    } else {
13        return 0;
14    }
15 }
16
17 void printPerfectNumbers(int n) {
18     printf("Perfect numbers between 1 and %d are:\n", n);
19     for (int i = 1; i <= n; i++) {
20         if (isPerfectNumber(i)) {
21             printf("%d ", i);
22         }
23     }
24     printf("\n");
25 }
26
27 int main() {
28     int n;
29
30     printf("AADARSH CHAUDHARY\nEnter a number: ");
31     scanf("%d", &n);
32
33     printPerfectNumbers(n);
34
35     return 0;
36 }
```

56. Write a C program to check whether a number is Strong number or not.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify.
- Code Editor:** File named "main.c". The code includes a factorial function and an isStrongNumber function. The isStrongNumber function calculates the sum of the factorials of the digits of a number and checks if it equals the original number. The main function prompts the user for a number and prints whether it is a strong number or not.
- Output Window:** Labeled "input". It displays the user input "AADARSH CHAUDHARY" and the program's output: "Enter a number: 145", and "145 is a Strong number."

```
1 #include <stdio.h>
2
3 int factorial(int num) {
4     if (num == 0 || num == 1) {
5         return 1;
6     } else {
7         return num * factorial(num - 1);
8     }
9 }
10
11 int isStrongNumber(int num) {
12     int originalNum = num;
13     int sum = 0;
14
15     while (num > 0) {
16         int digit = num % 10;
17         sum += factorial(digit);
18         num /= 10;
19     }
20
21     if (sum == originalNum) {
22         return 1;
23     } else {
24         return 0;
25     }
26 }
27
28 int main() {
29     int num;
30
31     printf("AADARSH CHAUDHARY\nEnter a number: ");
32     scanf("%d", &num);
33
34     if (isStrongNumber(num)) {
35         printf("%d is a Strong number.\n", num);
36     } else {
37         printf("%d is not a Strong number.\n", num);
38     }
39
40     return 0;
41 }
```

57. Write a C program to print all Strong numbers between 1 to n.

The screenshot shows a C IDE interface with a dark theme. The main window displays the code for a C program named 'main.c'. The code defines a factorial function, an isStrongNumber function, and a printStrongNumbers function. It then calls printStrongNumbers from the main function, which prints the strong numbers between 1 and the user-specified number (500 in the example). The terminal window below shows the execution of the program and its output.

```
main.c
1 #include <stdio.h>
2
3 int factorial(int num) {
4     if (num == 0 || num == 1) {
5         return 1;
6     } else {
7         return num * factorial(num - 1);
8     }
9 }
10
11 int isStrongNumber(int num) {
12     int originalNum = num;
13     int sum = 0;
14
15     while (num > 0) {
16         int digit = num % 10;
17         sum += factorial(digit);
18         num /= 10;
19     }
20
21     if (sum == originalNum) {
22         return 1;
23     } else {
24         return 0;
25     }
26 }
27
28 void printStrongNumbers(int n) {
29     printf("Strong numbers between 1 and %d are:\n", n);
30     for (int i = 1; i <= n; i++) {
31         if (isStrongNumber(i)) {
32             printf("%d ", i);
33         }
34     }
35     printf("\n");
36 }
37
38 int main() {
39     int n;
40
41     printf("AADARSH CHAUDHARY\nEnter a number: ");
42     scanf("%d", &n);
43
44     printStrongNumbers(n);
45
46     return 0;
47 }
```

```
AADARSH CHAUDHARY
Enter a number: 500
Strong numbers between 1 and 500 are:
1 2 145
```

58. Write a C program to print Fibonacci series up to n terms.

The screenshot shows a C IDE interface with a dark theme. The main window displays the code for a C program named 'main.c'. The code defines a printFibonacciSeries function, which prints the Fibonacci series up to n terms. It then calls printFibonacciSeries from the main function, which prints the series for the user-specified number (8 in the example). The terminal window below shows the execution of the program and its output.

```
main.c
1 #include <stdio.h>
2
3 void printFibonacciSeries(int n) {
4     int firstTerm = 0, secondTerm = 1;
5     int nextTerm;
6
7     printf("Fibonacci series up to %d terms:\n", n);
8
9     for (int i = 1; i <= n; i++) {
10        printf("%d ", firstTerm);
11        nextTerm = firstTerm + secondTerm;
12        firstTerm = secondTerm;
13        secondTerm = nextTerm;
14    }
15
16    printf("\n");
17 }
18
19 int main() {
20     int n;
21
22     printf("AADARSH CHAUDHARY\nEnter the number of terms: ");
23     scanf("%d", &n);
24
25     printFibonacciSeries(n);
26
27     return 0;
28 }
```

```
AADARSH CHAUDHARY
Enter the number of terms: 8
Fibonacci series up to 8 terms:
0 1 1 2 3 5 8 13
```

59. Write a C program to find one's complement of a binary number.

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.) and project management (Run, Debug, Stop, Share, Save, Beautify). The current file is "main.c".

```
1 #include <stdio.h>
2
3 void findOnesComplement(char binaryNumber[]) {
4     int i = 0;
5
6     // Finding the length of the binary number
7     while (binaryNumber[i] != '\0') {
8         i++;
9     }
10
11    // Performing one's complement by flipping each bit
12    for (int j = 0; j < i; j++) {
13        if (binaryNumber[j] == '0') {
14            binaryNumber[j] = '1';
15        } else if (binaryNumber[j] == '1') {
16            binaryNumber[j] = '0';
17        }
18    }
19
20    printf("One's complement of the binary number is: %s\n", binaryNumber);
21 }
22
23 int main() {
24     char binaryNumber[100];
25
26     printf("AADARSH CHAUDHARY\nEnter a binary number: ");
27     scanf("%s", binaryNumber);
28
29     findOnesComplement(binaryNumber);
30
31     return 0;
32 }
```

The output window below the editor shows the following interaction:

```
AADARSH CHAUDHARY
Enter a binary number: 010111
One's complement of the binary number is: 101000
```

60. Write a C program to find two's complement of a binary number.

The screenshot shows a code editor window with the following details:

- Title Bar:** Includes icons for file operations (New, Open, Save), run/debug mode (Run, Debug, Stop, Share, Save, Beautify).
- File:** main.c
- Code Content:** A C program named `main.c` that defines a function `findTwosComplement` and contains a `main` function.

```
1 #include <stdio.h>
2
3 void findTwosComplement(char binaryNumber[]) {
4     int i = 0;
5
6     // Finding the length of the binary number
7     while (binaryNumber[i] != '\0') {
8         i++;
9     }
10
11    // Finding the rightmost 1 in the binary number
12    int rightmostOne = -1;
13    for (int j = i - 1; j >= 0; j--) {
14        if (binaryNumber[j] == '1') {
15            rightmostOne = j;
16            break;
17        }
18    }
19
20    // Performing one's complement by flipping each bit after the rightmost 1
21    for (int j = rightmostOne - 1; j >= 0; j--) {
22        if (binaryNumber[j] == '0') {
23            binaryNumber[j] = '1';
24        } else if (binaryNumber[j] == '1') {
25            binaryNumber[j] = '0';
26        }
27    }
28
29    printf("Two's complement of the binary number is: %s\n", binaryNumber);
30 }
31
32 int main() {
33     char binaryNumber[100];
34
35     printf("AADARSH CHAUDHARY\nEnter a binary number: ");
36     scanf("%s", binaryNumber);
37
38     findTwosComplement(binaryNumber);
39
40     return 0;
41 }
```

- Output Panel:** Labeled "Input" at the top. It displays the program's output:

AADARSH CHAUDHARY
Enter a binary number: 101000
Two's complement of the binary number is: 011000

61. Write a C program to convert Binary to Octal number system.

The screenshot shows a C IDE interface with a toolbar at the top and a code editor below. The code editor has a dark theme and displays the following C program:

```
main.c
1 #include <stdio.h>
2 #include <math.h>
3
4 int convertBinaryToOctal(long long binaryNumber) {
5     int octalNumber = 0, decimalNumber = 0, i = 0;
6
7     // Converting binary to decimal
8     while (binaryNumber != 0) {
9         decimalNumber += (binaryNumber % 10) * pow(2, i);
10        ++i;
11        binaryNumber /= 10;
12    }
13
14    i = 1;
15
16    // Converting decimal to octal
17    while (decimalNumber != 0) {
18        octalNumber += (decimalNumber % 8) * i;
19        decimalNumber /= 8;
20        i *= 10;
21    }
22
23    return octalNumber;
24 }
25
26 int main() {
27     long long binaryNumber;
28
29     printf("AADARSH CHAUDHARY\nEnter a binary number: ");
30     scanf("%lld", &binaryNumber);
31
32     int octalNumber = convertBinaryToOctal(binaryNumber);
33
34     printf("The octal representation of the binary number is: %d\n", octalNumber);
35
36     return 0;
37 }
```

Below the code editor is a terminal window titled "input" showing the program's output:

```
AADARSH CHAUDHARY
Enter a binary number: 1010101
The octal representation of the binary number is: 125
```

62. Write a C program to convert Binary to Decimal number system.

The screenshot shows a C IDE interface with a toolbar at the top and a code editor below. The code editor has a dark theme and displays the following C program:

```
main.c
1 #include <stdio.h>
2 #include <math.h>
3
4 int convertBinaryToDecimal(long long binaryNumber) {
5     int decimalNumber = 0, i = 0;
6
7     // Converting binary to decimal
8     while (binaryNumber != 0) {
9         decimalNumber += (binaryNumber % 10) * pow(2, i);
10        ++i;
11        binaryNumber /= 10;
12    }
13
14    return decimalNumber;
15 }
16
17 int main() {
18     long long binaryNumber;
19
20     printf("AADARSH CHAUDHARY\nEnter a binary number: ");
21     scanf("%lld", &binaryNumber);
22
23     int decimalNumber = convertBinaryToDecimal(binaryNumber);
24
25     printf("The decimal representation of the binary number is: %d\n", decimalNumber);
26
27     return 0;
28 }
```

Below the code editor is a terminal window titled "input" showing the program's output:

```
AADARSH CHAUDHARY
Enter a binary number: 0101010
The decimal representation of the binary number is: 10
```

63. Write a C program to convert Binary to Hexadecimal number system.

The screenshot shows a code editor interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, and Beautify. The file tab shows "main.c". The code itself is a C program that converts a binary number to its hexadecimal representation. It uses two helper functions: `convertToHex` for digits 10-15 and `convertBinaryToHexadecimal` for the main conversion process. The output window at the bottom shows the program's interaction with the user, including prompts for input and the resulting output.

```
main.c
1 #include <stdio.h>
2 #include <math.h>
3
4 char convertToHex(int decimalNumber) {
5     if (decimalNumber >= 0 && decimalNumber <= 9) {
6         return (char)(decimalNumber + '0');
7     } else {
8         return (char)(decimalNumber - 10 + 'A');
9     }
10}
11
12 void convertBinaryToHexadecimal(long long binaryNumber) {
13     int decimalNumber = 0, i = 0;
14     char hexadecimalNumber[100];
15
16     // Converting binary to decimal
17     while (binaryNumber != 0) {
18         decimalNumber += (binaryNumber % 10) * pow(2, i);
19         ++i;
20         binaryNumber /= 10;
21     }
22
23     i = 0;
24
25     // Converting decimal to hexadecimal
26     while (decimalNumber != 0) {
27         int remainder = decimalNumber % 16;
28         hexadecimalNumber[i] = convertToHex(remainder);
29         decimalNumber /= 16;
30         ++i;
31     }
32
33     printf("The hexadecimal representation of the binary number is: ");
34     for (int j = i - 1; j >= 0; --j) {
35         printf("%c", hexadecimalNumber[j]);
36     }
37     printf("\n");
38 }
39
40 int main() {
41     long long binaryNumber;
42
43     printf("AADARSH CHAUDHARY\nEnter a binary number: ");
44     scanf("%lld", &binaryNumber);
45
46     convertBinaryToHexadecimal(binaryNumber);
47
48     return 0;
49 }
```

input

```
AADARSH CHAUDHARY
Enter a binary number: 110011
The hexadecimal representation of the binary number is: 33
```

64. Write a C program to convert Octal to Binary number system.

The screenshot shows a terminal window with the following content:

```
AADARSH CHAUDHARY
Enter an octal number: 248
The binary representation of the octal number is: 10101
```

65. Write a C program to convert Octal to Decimal number system.

The screenshot shows a terminal window with the following content:

```
AADARSH CHAUDHARY
Enter an octal number: 24
The decimal representation of the octal number is: 20

...Program finished with exit code 0
Press ENTER to exit console.
```

66. Write a C program to convert Octal to Hexadecimal number system.

The screenshot shows a code editor window with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, and Beautify.
- File:** main.c
- Code Content:** A C program that includes headers for stdio.h and math.h. It defines a main() function that prompts the user for an octal number, converts it to decimal, and then converts it to hexadecimal. The output is printed to the console.
- Output Panel:** Shows the execution results:
 - AADARSH CHAUDHARY
 - Enter an octal number: 64
 - The hexadecimal representation of the octal number is: 34
- Bottom Status:** Shows "input" and some small icons.

67. Write a C program to convert Decimal to Binary number system.

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.) and application controls (Run, Debug, Stop, Share, Save, Beautify). The file tab shows 'main.c'. The code area contains the following C program:

```
1 #include <stdio.h>
2
3 void decimalToBinary(int decimalNumber) {
4     int binaryNumber[32];
5     int i = 0;
6
7     while (decimalNumber > 0) {
8         binaryNumber[i] = decimalNumber % 2;
9         decimalNumber /= 2;
10        i++;
11    }
12
13    printf("The binary representation of the decimal number is: ");
14    for (int j = i - 1; j >= 0; j--) {
15        printf("%d", binaryNumber[j]);
16    }
17    printf("\n");
18 }
19
20 int main() {
21     int decimalNumber;
22
23     printf("AADARSH CHAUDHARY\nEnter a decimal number: ");
24     scanf("%d", &decimalNumber);
25
26     decimalToBinary(decimalNumber);
27
28     return 0;
29 }
```

Below the code area is a terminal window showing the execution of the program. The user has typed "AADARSH CHAUDHARY" followed by "Enter a decimal number: 24". The program then outputs "The binary representation of the decimal number is: 11000".

68. Write a C program to convert Decimal to Octal number system.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, and Beautify.
- File:** The file is named "main.c".
- Code Area:** Contains the following C code:

```
1 #include <stdio.h>
2
3 void decimalToOctal(int decimalNumber) {
4     int octalNumber[100];
5     int i = 0;
6
7     while (decimalNumber > 0) {
8         octalNumber[i] = decimalNumber % 8;
9         decimalNumber /= 8;
10        i++;
11    }
12
13    printf("The octal representation of the decimal number is: ");
14    for (int j = i - 1; j >= 0; j--) {
15        printf("%d", octalNumber[j]);
16    }
17    printf("\n");
18 }
19
20 int main() {
21     int decimalNumber;
22
23     printf("AADARSH CHAUDHARY\nEnter a decimal number: ");
24     scanf("%d", &decimalNumber);
25
26     decimalToOctal(decimalNumber);
27
28     return 0;
29 }
```
- Output Area:** Labeled "input" and displays the following text:

```
AADARSH CHAUDHARY
Enter a decimal number: 244
The octal representation of the decimal number is: 364
```

69. Write a C program to convert Decimal to Hexadecimal number system.

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.) and buttons for Run, Debug, Stop, Share, Save, and Beautify. The file tab shows "main.c". The code itself is a C program that defines a function to convert a decimal number to its hexadecimal representation and prints it. It includes a main() function that reads a decimal number from the user and calls the conversion function. The output window at the bottom shows the program's execution and the resulting hex output.

```
1 #include <stdio.h>
2
3 void decimalToHexadecimal(int decimalNumber) {
4     char hexadecimalNumber[100];
5     int i = 0;
6
7     while (decimalNumber > 0) {
8         int remainder = decimalNumber % 16;
9         if (remainder < 10) {
10             hexadecimalNumber[i] = remainder + '0';
11         } else {
12             hexadecimalNumber[i] = remainder + 55;
13         }
14         decimalNumber /= 16;
15         i++;
16     }
17
18     printf("The hexadecimal representation of the decimal number is: ");
19     for (int j = i - 1; j >= 0; j--) {
20         printf("%c", hexadecimalNumber[j]);
21     }
22     printf("\n");
23 }
24
25 int main() {
26     int decimalNumber;
27
28     printf("AADARSH CHAUDHARY\nEnter a decimal number: ");
29     scanf("%d", &decimalNumber);
30
31     decimalToHexadecimal(decimalNumber);
32
33     return 0;
34 }
```

AADARSH CHAUDHARY
Enter a decimal number: 545
The hexadecimal representation of the decimal number is: 221

70. Write a C program to convert Hexadecimal to Binary number system.

The screenshot shows a code editor window with the following details:

- Title Bar:** Includes icons for file operations (New, Open, Save), a "Run" button, a "Debug" button, a "Stop" button, a "Share" button, a "Save" button, a "Beautify" button, and a download icon.
- File Name:** main.c
- Code Content:** A C program that reads a hexadecimal number from the user, converts it to decimal, and then converts it to binary. The code uses character arrays for the hex string and loops through each character to calculate the decimal value. It then uses a while loop to repeatedly divide the decimal number by 2 to find the binary digits.

```
1 #include <stdio.h>
2
3 int main() {
4     char hexadecimalNumber[20];
5     long int decimalNumber = 0;
6     int i = 0;
7
8     printf("AADARSH CHAUDHARY\nEnter a hexadecimal number: ");
9     scanf("%s", hexadecimalNumber);
10
11    // Convert hexadecimal to decimal
12    for (i = 0; hexadecimalNumber[i] != '\0'; i++) {
13        if (hexadecimalNumber[i] >= '0' && hexadecimalNumber[i] <= '9') {
14            decimalNumber = decimalNumber * 16 + (hexadecimalNumber[i] - '0');
15        }
16        else if (hexadecimalNumber[i] >= 'A' && hexadecimalNumber[i] <= 'F') {
17            decimalNumber = decimalNumber * 16 + (hexadecimalNumber[i] - 'A' + 10);
18        }
19        else if (hexadecimalNumber[i] >= 'a' && hexadecimalNumber[i] <= 'f') {
20            decimalNumber = decimalNumber * 16 + (hexadecimalNumber[i] - 'a' + 10);
21        }
22        else {
23            printf("Invalid hexadecimal number!\n");
24            return 0;
25        }
26    }
27
28    // Convert decimal to binary
29    long int binaryNumber = 0, remainder, base = 1;
30    while (decimalNumber > 0) {
31        remainder = decimalNumber % 2;
32        binaryNumber = binaryNumber + remainder * base;
33        decimalNumber = decimalNumber / 2;
34        base = base * 10;
35    }
36
37    printf("The binary representation of the hexadecimal number is: %ld\n", binaryNumber);
38
39    return 0;
40 }
```

- Output Window:** Shows the program's output:

AADARSH CHAUDHARY
Enter a hexadecimal number: 24
The binary representation of the hexadecimal number is: 100100

71. Write a C program to convert Hexadecimal to Octal number system.

The screenshot shows a code editor window with a dark theme. The file is named 'main.c'. The code implements two functions: 'hexToDecimal' and 'decimalToOctal', and a main function 'main'. The 'hexToDecimal' function converts a hexadecimal string to a decimal integer. It first finds the length of the input string. Then, it iterates through each character, converts it to its decimal value (considering if it's an uppercase or lowercase letter), and adds it to the result after multiplying by 16 raised to the power of the current position's index. The 'decimalToOctal' function converts a decimal integer to an octal string. It repeatedly divides the decimal number by 8, stores the remainder, and then multiplies the quotient by 10. This process continues until the decimal number becomes 0. The 'main' function prompts the user for a hexadecimal input, calls 'hexToDecimal' to get the decimal value, calls 'decimalToOctal' to get the octal representation, and then prints the result.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 // Function to convert hexadecimal to decimal
5 int hexToDecimal(char hex[]) {
6     int decimal = 0, i = 0, val, len;
7
8     // Find the length of the hexadecimal number
9     for (len = 0; hex[len] != '\0'; len++);
10
11    // Convert each digit to decimal
12    for (i = 0; hex[i] != '\0'; i++, len--) {
13        if (hex[i] >= '0' && hex[i] <= '9') {
14            val = hex[i] - '0';
15        } else if (hex[i] >= 'A' && hex[i] <= 'F') {
16            val = hex[i] - 'A' + 10;
17        } else if (hex[i] >= 'a' && hex[i] <= 'f') {
18            val = hex[i] - 'a' + 10;
19        }
20        decimal += val * pow(16, len - 1);
21    }
22
23    return decimal;
24 }
25
26 // Function to convert decimal to octal
27 long decimalToOctal(int decimal) {
28     long octal = 0;
29     int i = 1;
30
31     while (decimal != 0) {
32         octal += (decimal % 8) * i;
33         decimal /= 8;
34         i *= 10;
35     }
36
37     return octal;
38 }
39
40 int main() {
41     char hex[20];
42
43     printf("AADARSH CHAUDHARY\nEnter a hexadecimal number: ");
44     scanf("%s", hex);
45
46     int decimal = hexToDecimal(hex);
47     long octal = decimalToOctal(decimal);
48
49     printf("The octal representation of the hexadecimal number is: %lo\n", octal);
50
51     return 0;
52 }
```

AADARSH CHAUDHARY
Enter a hexadecimal number: 48
The octal representation of the hexadecimal number is: 156

72. Write a C program to convert Hexadecimal to Decimal number system.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor:** The file "main.c" contains the following C code:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 // Function to convert hexadecimal to decimal
5 int hexToDecimal(char hex[]) {
6     int decimal = 0, base = 1, len = 0, i;
7
8     // Calculating the length of the hexadecimal number
9     while (hex[len] != '\0') {
10         len++;
11     }
12
13     // Converting each digit of the hexadecimal number to decimal
14     for (i = len - 1; i >= 0; i--) {
15         if (hex[i] >= '0' && hex[i] <= '9') {
16             decimal += (hex[i] - '0') * base;
17         } else if (hex[i] >= 'A' && hex[i] <= 'F') {
18             decimal += (hex[i] - 'A' + 10) * base;
19         } else if (hex[i] >= 'a' && hex[i] <= 'f') {
20             decimal += (hex[i] - 'a' + 10) * base;
21         }
22         base *= 16;
23     }
24
25     return decimal;
26 }
27
28 int main() {
29     char hex[20];
30
31     printf("AADARSH CHAUDHARY\nEnter a hexadecimal number: ");
32     scanf("%s", hex);
33
34     int decimal = hexToDecimal(hex);
35
36     printf("The decimal representation of the hexadecimal number is: %d\n", decimal);
37
38     return 0;
39 }
```
- Output Window:** Shows the program's execution:

```
AADARSH CHAUDHARY
Enter a hexadecimal number: 56
The decimal representation of the hexadecimal number is: 86
```

Pattern Exercises

1. Star pattern programs - Write a C program to print the given star patterns.

Pyramid Star Pattern

The screenshot shows a code editor interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, and Beautify. The file tab shows "main.c". The code itself is a C program that prints a pyramid star pattern based on user input for the number of rows. The output window shows the program's execution and the resulting star pattern.

```
1 #include <stdio.h>
2
3 int main() {
4     int rows, i, j;
5
6     printf("ADARSH CHAUDHARY\nEnter the number of rows: ");
7     scanf("%d", &rows);
8
9     for (i = 1; i <= rows; i++) {
10        for (j = 1; j <= rows - i; j++) {
11            printf(" ");
12        }
13        for (j = 1; j <= 2 * i - 1; j++) {
14            printf("*");
15        }
16        printf("\n");
17    }
18
19    return 0;
20 }
```

ADARSH CHAUDHARY
Enter the number of rows: 5

```
*  
***  
*****  
*****  
*****  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Hollow Pyramid Star Pattern

The screenshot shows a code editor interface with a dark theme. At the top, there are several buttons: Run (green), Debug (blue), Stop (red), Share (orange), Save (blue), and Beautify (light blue). The file tab is labeled "main.c". The code itself is a C program that prints a hollow pyramid star pattern. It includes a header file inclusion, a main function, a printf for input, a scanf for input, and two nested loops for printing stars and spaces. The output window below shows the program's execution and the resulting hollow pyramid pattern.

```
1 #include <stdio.h>
2
3 int main() {
4     int rows, i, j;
5
6     printf("AADARSH CHAUDHARY\nEnter the number of rows: ");
7     scanf("%d", &rows);
8
9     for (i = 1; i <= rows; i++) {
10        for (j = 1; j <= rows - i; j++) {
11            printf(" ");
12        }
13        for (j = 1; j <= 2 * i - 1; j++) {
14            if (j == 1 || j == 2 * i - 1 || i == rows) {
15                printf("*");
16            } else {
17                printf(" ");
18            }
19        }
20        printf("\n");
21    }
22
23    return 0;
24 }
```

AADARSH CHAUDHARY
Enter the number of rows: 5

```
*  
* *  
*   *  
*   *  
*****
```

Inverted Pyramid Star Pattern

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.) and a set of buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The current file is "main.c", which contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int rows, i, j;
5
6     printf("AADARSH CHAUDHARY\nEnter the number of rows: ");
7     scanf("%d", &rows);
8
9     for (i = rows; i >= 1; i--) {
10        for (j = 1; j <= rows - i; j++) {
11            printf(" ");
12        }
13        for (j = 1; j <= 2 * i - 1; j++) {
14            printf("*");
15        }
16        printf("\n");
17    }
18
19    return 0;
20 }
21
```

Below the code editor is a terminal window showing the execution of the program. The user has typed "AADARSH CHAUDHARY" followed by "Enter the number of rows: 5". The program then prints an inverted pyramid of stars:

```
AADARSH CHAUDHARY
Enter the number of rows: 5
*****
 ****
  ***
   *

```

At the bottom of the terminal window, the message "...Program finished with exit code 0" is displayed, followed by "Press ENTER to exit console."

Hollow Inverted Pyramid Star Pattern

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for Run, Debug, Stop, Share, Save, and Beautify. The file tab shows "main.c". The code in the editor is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int rows = 5;
5
6     printf("AADARSH CHAUDHARY\n");
7     for (int i = rows; i >= 1; i--) {
8         for (int j = rows; j > i; j--) {
9             printf(" ");
10        }
11        for (int k = 1; k <= 2 * i - 1; k++) {
12            if (k == 1 || k == 2 * i - 1 || i == rows) {
13                printf("*");
14            } else {
15                printf(" ");
16            }
17        }
18        printf("\n");
19    }
20
21    return 0;
22 }
```

The output window below the editor shows the execution results:

```
AADARSH CHAUDHARY
*****
*   *
*   *
* *
*
...Program finished with exit code 0
Press ENTER to exit console.
```

Half Diamond Star Pattern

The screenshot shows a code editor interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save), run/debug/stop, share, save, beautify, and download.
- File:** The current file is "main.c".
- Code:** The C code prints a half diamond star pattern. It starts with a header, prompts for rows, and then uses nested loops to print stars. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int rows, i, j;
5
6     printf("AADARSH CHAUDHARY\nEnter the number of rows: ");
7     scanf("%d", &rows);
8
9     for (i = 1; i <= rows; i++) {
10        for (j = 1; j <= i; j++) {
11            printf("*");
12        }
13        printf("\n");
14    }
15
16    for (i = rows - 1; i >= 1; i--) {
17        for (j = 1; j <= i; j++) {
18            printf("*");
19        }
20        printf("\n");
21    }
22
23    return 0;
24 }
```

- Output:** The terminal window shows the program's output:

```
AADARSH CHAUDHARY
Enter the number of rows: 5
*
**
***
****
*****
****
***
**
*
```

Mirrored Half Diamond Star Pattern

The screenshot shows a C code editor interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to C. The code file is named main.c. The code itself is as follows:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("AADARSH CHAUDHARY\n");
6     int i, j, N;
7     int star, spaces;
8
9     /* Input number of columns to print from user */
10    printf("Enter number of columns : ");
11    scanf("%d", &N);
12
13
14    spaces = N-1; star = 1;
15
16    /* Iterate through rows */
17    for(i=1; i<N*2; i++)
18    {
19        /* Print leading spaces */
20        for(j=1; j<=spaces; j++)
21            printf(" ");
22
23        /* Print stars */
24        for(j=1; j<=star; j++)
25            printf("*");
26
27        /* Move to next line */
28        printf("\n");
29
30        if(i < N)
31        {star++; spaces--;}
32        else
33        {star--; spaces++;}
34    }
35
36
37    return 0;
38 }
```

The output window below the editor shows the program's execution. It starts with the author's name, "AADARSH CHAUDHARY", followed by an input prompt "Enter number of columns : 5". The program then prints a mirrored half diamond star pattern with 5 columns:

```
AADARSH CHAUDHARY
Enter number of columns : 5
    *
   **
  ***
 ****
*****
 ****
  ***
   **
    *
```

2. Number pattern programs - Write a C program to print the given number pattern

Patterns Square number patterns

11111

11111

11111

11111

11111

11111

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.) and buttons for Run, Debug, Stop, Share, Save, Beautify, and Download. The current file is "main.c". The code itself is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int rows = 5;
5     int cols = 5;
6
7     printf("AADARSH CHAUDHARY\n");
8     for (int i = 1; i <= rows; i++) {
9         for (int j = 1; j <= cols; j++) {
10            printf("1");
11        }
12        printf("\n");
13    }
14
15    return 0;
16 }
17
```

Below the code editor is a terminal window titled "input" showing the output of the program. The output consists of the string "AADARSH CHAUDHARY" followed by five lines of the number pattern "11111".

```
AADARSH CHAUDHARY
11111
11111
11111
11111
11111
```

Number pattern

11111

00000

11111

00000

11111

The screenshot shows a code editor interface with a dark theme. At the top is a toolbar with icons for file operations (New, Open, Save, etc.) and application functions (Run, Debug, Stop, Share, Save, Beautify). The main window has a tab labeled "main.c". The code itself is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int rows = 5;
5     int cols = 5;
6
7     printf("AADARSH CHAUDHARY\n");
8     for (int i = 1; i <= rows; i++) {
9         for (int j = 1; j <= cols; j++) {
10            if (i % 2 == 1) {
11                printf("1");
12            } else {
13                printf("0");
14            }
15        }
16        printf("\n");
17    }
18
19    return 0;
20 }
```

Below the code editor is a terminal window titled "input" showing the output of the program. The output consists of five lines of alternating "1"s and "0"s, followed by the message "...Program finished with exit code 0 Press ENTER to exit console.".

```
AADARSH CHAUDHARY
11111
00000
11111
00000
11111

...Program finished with exit code 0
Press ENTER to exit console.
```

Number pattern

01010

01010

01010

01010

01010

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, Share, Beautify) and a download button. Below the toolbar, the file name "main.c" is displayed. The code itself is a C program that prints a pattern of numbers. It starts by including the standard input-output library. It defines variables for rows and columns, both set to 5. It then prints a header "AADARSH CHAUDHARY\n". It uses nested loops to iterate through each position in a 5x5 grid. For each position, it checks if the column index j is even (using the condition $j \% 2 == 0$). If true, it prints a "1"; otherwise, it prints a "0". After printing all positions in a row, it prints a new line character "\n". Finally, it returns 0 from the main function.

```
1 #include <stdio.h>
2
3 int main() {
4     int rows = 5;
5     int cols = 5;
6
7     printf("AADARSH CHAUDHARY\n");
8     for (int i = 1; i <= rows; i++) {
9         for (int j = 1; j <= cols; j++) {
10            if (j % 2 == 0) {
11                printf("1");
12            } else {
13                printf("0");
14            }
15        }
16        printf("\n");
17    }
18
19    return 0;
20 }
21
```

AADARSH CHAUDHARY

01010

01010

01010

01010

01010

Number pattern

11111

10001

10001

10001

11111

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.) and buttons for Run, Debug, Stop, Share, Save, Beautify, and Download. The current file is "main.c". The code itself is a C program that prints a 5x5 grid of numbers. It starts by including stdio.h, defining rows and cols as 5, and printing the string "AADARSH CHAUDHARY\n". It then uses nested loops to iterate through a 5x5 grid. If the row index (i) or column index (j) is 1, or if i equals rows or j equals cols, it prints a '1'. Otherwise, it prints a '0'. Finally, it prints a new line character. The output window at the bottom shows the printed pattern:

```
AADARSH CHAUDHARY
11111
10001
10001
10001
11111
```

Number pattern

11111

11111

11011

11111

11111

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.) and application-specific functions (Run, Debug, Stop, Share, Beautify). The main window displays a file named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int rows = 5;
5     int cols = 5;
6
7     printf("AADARSH CHAUDHARY\n");
8     for (int i = 1; i <= rows; i++) {
9         for (int j = 1; j <= cols; j++) {
10            if (i == 3 && j == 3) {
11                printf("0");
12            } else {
13                printf("1");
14            }
15        }
16        printf("\n");
17    }
18
19    return 0;
20 }
```

Below the code editor is a terminal window titled "input" showing the output of the program. The output consists of five lines of text: "AADARSH CHAUDHARY", followed by five "11111" patterns, and then a final "11111".

Number pattern

10101

01010

10101

01010

10101

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (down arrow, up arrow), run (green play button), debug (blue circle), stop (red square), share (orange square), save (blue square with H), beautify (blue square with brackets), and a download icon.

The main window displays a file named "main.c". The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int rows = 5;
5     int cols = 5;
6
7     printf("AADARSH CHAUDHARY\n");
8     for (int i = 1; i <= rows; i++) {
9         for (int j = 1; j <= cols; j++) {
10            if ((i + j) % 2 == 0) {
11                printf("1");
12            } else {
13                printf("0");
14            }
15        }
16        printf("\n");
17    }
18
19    return 0;
20 }
```

Below the code editor is a terminal window with the title "input". It shows the output of the program:

```
AADARSH CHAUDHARY
10101
01010
10101
01010
10101
```

If...Else Exercises

1. Write a C program to find maximum between two numbers.

The screenshot shows a C IDE interface with a code editor and a terminal window. The code editor displays a file named 'main.c' containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int num1, num2;
5
6     printf("AADARSH CHAUDHARY\nEnter the first number: ");
7     scanf("%d", &num1);
8
9     printf("Enter the second number: ");
10    scanf("%d", &num2);
11
12    if (num1 > num2) {
13        printf("The maximum number is: %d\n", num1);
14    } else {
15        printf("The maximum number is: %d\n", num2);
16    }
17
18    return 0;
19 }
20
```

The terminal window below shows the program's output:

```
AADARSH CHAUDHARY
Enter the first number: 50
Enter the second number: 40
The maximum number is: 50

...Program finished with exit code 0
Press ENTER to exit console.
```

2. Write a C program to find maximum between three numbers.

The screenshot shows a code editor window for a C program named 'main.c'. The code uses standard input/output functions and an if-else conditional statement to determine the maximum of three integers. The output window below shows the program's execution and the results of running it with specific inputs.

```
1 #include <stdio.h>
2
3 int main() {
4     int num1, num2, num3, max;
5
6     printf("AADARSH CHAUDHARY\nEnter the first number: ");
7     scanf("%d", &num1);
8
9     printf("Enter the second number: ");
10    scanf("%d", &num2);
11
12    printf("Enter the third number: ");
13    scanf("%d", &num3);
14
15    if (num1 >= num2 && num1 >= num3) {
16        max = num1;
17    } else if (num2 >= num1 && num2 >= num3) {
18        max = num2;
19    } else {
20        max = num3;
21    }
22
23    printf("The maximum number is: %d\n", max);
24
25    return 0;
26 }
```

AADARSH CHAUDHARY
Enter the first number: 20
Enter the second number: 10
Enter the third number: 30
The maximum number is: 30

3. Write a C program to check whether a number is negative, positive or zero.

The screenshot shows a C programming environment with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save), Run, Debug, Stop, Share, and Beautify.
- Code Editor:** File name: main.c. The code checks if a number is positive, negative, or zero using an if-else-if-else structure.

```
1 #include <stdio.h>
2
3 int main() {
4     int num;
5
6     printf("AADARSH CHAUDHARY\nEnter a number: ");
7     scanf("%d", &num);
8
9     if (num > 0) {
10         printf("The number is positive.\n");
11     } else if (num < 0) {
12         printf("The number is negative.\n");
13     } else {
14         printf("The number is zero.\n");
15     }
16
17     return 0;
18 }
19
```

- Output Console:** Shows the program's output. The user enters "20" as input, and the program outputs "The number is positive." followed by the standard completion message.

```
AADARSH CHAUDHARY
Enter a number: 20
The number is positive.

...Program finished with exit code 0
Press ENTER to exit console.
```

4. Write a C program to check whether a number is divisible by 5 and 11 or not.

The screenshot shows a C IDE interface with the following components:

- Toolbar:** Includes icons for file operations (New, Open, Save), Run, Debug, Stop, Share, and Beautify.
- Code Editor:** A dark-themed editor with syntax highlighting for C code. The file is named "main.c". The code checks if a user-specified number is divisible by both 5 and 11.
- Output Console:** Labeled "input" at the top. It displays the program's output:
 - User input: AADARSH CHAUDHARY
 - User input: Enter a number: 110
 - Program output: The number is divisible by both 5 and 11.
 - Completion message: ...Program finished with exit code 0
 - Completion message: Press ENTER to exit console.

5. Write a C program to check whether a number is even or odd.

The screenshot shows a C code editor interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save), run (Run, Debug, Stop, Share, Save, Beautify).
- Code Area:** File name: main.c. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int num;
5
6     printf("AADARSH CHAUDHARY\nEnter a number: ");
7     scanf("%d", &num);
8
9     if (num % 2 == 0) {
10         printf("The number is even.\n");
11     } else {
12         printf("The number is odd.\n");
13     }
14
15     return 0;
16 }
17
```

Output Console:

```
AADARSH CHAUDHARY
Enter a number: 24
The number is even.

...Program finished with exit code 0
Press ENTER to exit console.
```

6. Write a C program to check whether a year is leap year or not.

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.), a 'Run' button, a 'Debug' button, a 'Stop' button, a 'Share' button, a 'Save' button, a 'Beautify' button, and a download icon. Below the toolbar, the file name 'main.c' is displayed. The code itself is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int year;
5
6     printf("AADARSH CHAUDHARY\nEnter a year: ");
7     scanf("%d", &year);
8
9     if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
10        printf("The year is a leap year.\n");
11    } else {
12        printf("The year is not a leap year.\n");
13    }
14
15    return 0;
16 }
17
```

Below the code editor is a terminal window titled 'input'. It displays the following output:

```
AADARSH CHAUDHARY
Enter a year: 2024
The year is a leap year.

...Program finished with exit code 0
Press ENTER to exit console.
```

7. Write a C program to check whether a character is alphabet or not.

The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with various icons: a file icon, a save icon, a run icon (highlighted in green), a debug icon, a stop icon, a share icon, a save icon, a beautify icon, and a download icon. Below the toolbar, the file name "main.c" is displayed. The code itself is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     char ch;
5
6     printf("AADARSH CHAUDHARY\nEnter a character: ");
7     scanf(" %c", &ch);
8
9     if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {
10        printf("The character is an alphabet.\n");
11    } else {
12        printf("The character is not an alphabet.\n");
13    }
14
15    return 0;
16 }
17
```

Below the code editor is a terminal window titled "input". It displays the following output:

```
AADARSH CHAUDHARY
Enter a character: A
The character is an alphabet.

...Program finished with exit code 0
Press ENTER to exit console.
```

8. Write a C program to input any alphabet and check whether it is vowel or consonant.

The screenshot shows a C IDE interface with the following components:

- Toolbar:** Includes icons for file operations (New, Open, Save), run/debug/stop, share, save, and beautify.
- Code Editor:** The file "main.c" contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     char ch;
5
6     printf("AADARSH CHAUDHARY\nEnter an alphabet: ");
7     scanf(" %c", &ch);
8
9     if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
10    ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
11         printf("The alphabet is a vowel.\n");
12     } else {
13         printf("The alphabet is a consonant.\n");
14     }
15
16     return 0;
17 }
18
```
- Output Console:** Displays the program's output:

```
AADARSH CHAUDHARY
Enter an alphabet: U
The alphabet is a vowel.

...Program finished with exit code 0
Press ENTER to exit console.
```

9. Write a C program to input any character and check whether it is alphabet, digit or special character.

The screenshot shows a C IDE interface with the following components:

- Toolbar:** Includes icons for file operations (New, Open, Save), Run, Debug, Stop, Share, Save, and Beautify.
- Code Editor:** Displays the file "main.c" containing the provided C code.
- Output Console:** Shows the program's output, including the prompt "Enter a character: @" and the response "The character is a special character.".
- Status Bar:** Shows "input" above the console area.

```
1 #include <stdio.h>
2
3 int main() {
4     char ch;
5
6     printf("AADARSH CHAUDHARY\nEnter a character: ");
7     scanf(" %c", &ch);
8
9     if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {
10         printf("The character is an alphabet.\n");
11     } else if (ch >= '0' && ch <= '9') {
12         printf("The character is a digit.\n");
13     } else {
14         printf("The character is a special character.\n");
15     }
16
17     return 0;
18 }
```

AADARSH CHAUDHARY
Enter a character: @
The character is a special character.

...Program finished with exit code 0
Press ENTER to exit console.

10. Write a C program to check whether a character is uppercase or lowercase alphabet.

The screenshot shows a C IDE interface with the following components:

- Toolbar:** Includes icons for file operations (New, Open, Save), Run, Debug, Stop, Share, Save, Beautify, and Download.
- Code Editor:** The file "main.c" contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     char ch;
5
6     printf("AADARSH CHAUDHARY\nEnter a character: ");
7     scanf(" %c", &ch);
8
9     if (ch >= 'a' && ch <= 'z') {
10         printf("The character is a lowercase alphabet.\n");
11     } else if (ch >= 'A' && ch <= 'Z') {
12         printf("The character is an uppercase alphabet.\n");
13     } else {
14         printf("The character is not an alphabet.\n");
15     }
16
17     return 0;
18 }
```
- Output Console:** Displays the program's output:

```
AADARSH CHAUDHARY
Enter a character: a
The character is a lowercase alphabet.

...Program finished with exit code 0
Press ENTER to exit console.
```

11. Write a C program to input week number and print week day.

The screenshot shows a code editor interface with a dark theme. At the top, there are several icons: a file icon, a run icon, a debug icon, a stop icon, a share icon, a save icon, a beautify icon, and a download icon. To the right of these is a "Language" dropdown set to "C" and two small status icons.

The main window displays a C program named "main.c". The code uses a switch statement to map integers 1 through 7 to their corresponding days of the week: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. It includes error handling for invalid inputs.

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     int weekNumber;
5
6     printf("AADARSH CHAUDHARY\nEnter a week number (1-7): ");
7     scanf("%d", &weekNumber);
8
9     if (weekNumber == 1) {
10         printf("The weekday corresponding to week number 1 is Sunday.\n");
11     } else if (weekNumber == 2) {
12         printf("The weekday corresponding to week number 2 is Monday.\n");
13     } else if (weekNumber == 3) {
14         printf("The weekday corresponding to week number 3 is Tuesday.\n");
15     } else if (weekNumber == 4) {
16         printf("The weekday corresponding to week number 4 is Wednesday.\n");
17     } else if (weekNumber == 5) {
18         printf("The weekday corresponding to week number 5 is Thursday.\n");
19     } else if (weekNumber == 6) {
20         printf("The weekday corresponding to week number 6 is Friday.\n");
21     } else if (weekNumber == 7) {
22         printf("The weekday corresponding to week number 7 is Saturday.\n");
23     } else {
24         printf("Invalid week number. Please enter a number between 1 and 7.\n");
25     }
26
27     return 0;
28 }
```

Below the code editor is a terminal window titled "input". It shows the program's output after running it and entering the number 6 as input. The output reads:

```
AADARSH CHAUDHARY
Enter a week number (1-7): 6
The weekday corresponding to week number 6 is Friday.
```

12. Write a C program to input month number and print number of days in that month.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, and Beautify.
- Language Selection:** Set to C.
- Code Editor:** File named "main.c" containing the provided C code.
- Output Window:** Labeled "input" at the top, showing the program's execution. It displays:
 - User input: "AADARSH CHAUDHARY
 - User input: "Enter a month number (1-12): 4"
 - Program output: "April has 30 days."
 - Completion message: "...Program finished with exit code 0
 - Instruction: "Press ENTER to exit console."

```
1 #include <stdio.h>
2
3 int main() {
4     int monthNumber;
5
6     printf("AADARSH CHAUDHARY\nEnter a month number (1-12): ");
7     scanf("%d", &monthNumber);
8
9     if (monthNumber == 1) {
10         printf("January has 31 days.\n");
11     } else if (monthNumber == 2) {
12         printf("February has either 28 or 29 days, depending on the year.\n");
13     } else if (monthNumber == 3) {
14         printf("March has 31 days.\n");
15     } else if (monthNumber == 4) {
16         printf("April has 30 days.\n");
17     } else if (monthNumber == 5) {
18         printf("May has 31 days.\n");
19     } else if (monthNumber == 6) {
20         printf("June has 30 days.\n");
21     } else if (monthNumber == 7) {
22         printf("July has 31 days.\n");
23     } else if (monthNumber == 8) {
24         printf("August has 31 days.\n");
25     } else if (monthNumber == 9) {
26         printf("September has 30 days.\n");
27     } else if (monthNumber == 10) {
28         printf("October has 31 days.\n");
29     } else if (monthNumber == 11) {
30         printf("November has 30 days.\n");
31     } else if (monthNumber == 12) {
32         printf("December has 31 days.\n");
33     } else {
34         printf("Invalid month number. Please enter a number between 1 and 12.\n");
35     }
36
37     return 0;
38 }
```

13. Write a C program to count total number of notes in given amount.

The screenshot shows a code editor interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to C. The main window displays a C program named 'main.c'. The code uses nested loops and if statements to calculate the number of notes required for an input amount. It counts from 2000 down to 2 rupee coins. The output window at the bottom shows the program's execution and the result for an input of 300.

```
#include <stdio.h>
int main() {
    int amount, notes = 0;
    printf("AADARSH CHAUDHARY\nEnter the amount: ");
    scanf("%d", &amount);

    // Counting number of 2000 rupee notes
    if (amount >= 2000) {
        notes += amount / 2000;
        amount %= 2000;
    }

    // Counting number of 500 rupee notes
    if (amount >= 500) {
        notes += amount / 500;
        amount %= 500;
    }

    // Counting number of 200 rupee notes
    if (amount >= 200) {
        notes += amount / 200;
        amount %= 200;
    }

    // Counting number of 100 rupee notes
    if (amount >= 100) {
        notes += amount / 100;
        amount %= 100;
    }

    // Counting number of 50 rupee notes
    if (amount >= 50) {
        notes += amount / 50;
        amount %= 50;

    }

    // Counting number of 20 rupee notes
    if (amount >= 20) {
        notes += amount / 20;
        amount %= 20;
    }

    // Counting number of 10 rupee notes
    if (amount >= 10) {
        notes += amount / 10;
        amount %= 10;
    }

    // Counting number of 5 rupee notes
    if (amount >= 5) {
        notes += amount / 5;
        amount %= 5;
    }

    // Counting number of 2 rupee coins
    if (amount >= 2) {
        notes += amount / 2;
        amount %= 2;
    }

    // Counting number of 1 rupee coins
    if (amount >= 1) {
        notes += amount;
    }

    printf("The total number of notes is: %d\n", notes);
    return 0;
}
```

AADARSH CHAUDHARY
Enter the amount: 300
The total number of notes is: 2

14. Write a C program to input month number and print number of days in that month.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, and Beautify.
- Language:** Set to C.
- Code Editor:** File named "main.c" containing the provided C code.
- Output Window:** Labeled "input" at the top. It displays the program's output: "AADARSH CHAUDHARY", "Enter the month number: 5", and "This month has 31 days."

```
#include <stdio.h>
int main() {
    int month;
    printf("AADARSH CHAUDHARY\nEnter the month number: ");
    scanf("%d", &month);
    if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12) {
        printf("This month has 31 days.\n");
    } else if (month == 4 || month == 6 || month == 9 || month == 11) {
        printf("This month has 30 days.\n");
    } else if (month == 2) {
        printf("This month has 28 or 29 days.\n");
    } else {
        printf("Invalid month number.\n");
    }
    return 0;
}
```

15. Write a C program to count total number of notes in given amount.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, and Beautify.
- Language:** Set to C.
- Code Editor:** File named "main.c" containing the provided C code.
- Output Window:** Labeled "input" at the top. It displays the program's output: "The total number of notes:", followed by a list of note counts: "1 note of 2000", "1 note of 500", "1 note of 200", "1 note of 50", and "1 note of 20".

```
#include <stdio.h>
int main() {
    int amount;
    int notes[7] = {2000, 500, 200, 100, 50, 20, 10};
    int count[7] = {0};

    printf("AADARSH CHAUDHARY\nEnter the amount: ");
    scanf("%d", &amount);

    for (int i = 0; i < 7; i++) {
        if (amount >= notes[i]) {
            count[i] = amount / notes[i];
            amount %= notes[i];
        }
    }

    printf("The total number of notes:\n");
    for (int i = 0; i < 7; i++) {
        if (count[i] > 0) {
            printf("%d note of %d\n", count[i], notes[i]);
        }
    }
    return 0;
}
```

16. Write a C program to input angles of a triangle and check whether triangle is valid or not.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save), run/debug/stop, share, beautify, and download.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int angle1, angle2, angle3;
5
6     printf("AADARSH CHAUDHARY\nEnter the three angles of the triangle: ");
7     scanf("%d %d %d", &angle1, &angle2, &angle3);
8
9     if (angle1 + angle2 + angle3 == 180) {
10         printf("The triangle is valid!\n");
11     } else {
12         printf("The triangle is not valid.\n");
13     }
14
15     return 0;
16 }
17
```
- Output Window:** Labeled "input" at the top. It displays the program's output:

```
AADARSH CHAUDHARY
Enter the three angles of the triangle: 60 60 60
The triangle is valid!

...Program finished with exit code 0
Press ENTER to exit console.
```

17. Write a C program to input all sides of a triangle and check whether triangle is valid or not.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save), run/debug/stop, share, beautify, and download. Language dropdown set to C.
- Code Editor:** File named "main.c" containing the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     int side1, side2, side3;
5
6     printf("AADARSH CHAUDHARY\nEnter the lengths of all three sides of the triangle: ");
7     scanf("%d %d %d", &side1, &side2, &side3);
8
9     if (side1 + side2 > side3 && side2 + side3 > side1 && side1 + side3 > side2) {
10         printf("The triangle is valid!\n");
11     } else {
12         printf("The triangle is not valid.\n");
13     }
14
15     return 0;
16 }
17
```
- Output Window:** Labeled "input" at the top. It displays the program's output:

```
AADARSH CHAUDHARY
Enter the lengths of all three sides of the triangle: 20 20 20
The triangle is valid!

...Program finished with exit code 0
Press ENTER to exit console.
```

18. Write a C program to check whether the triangle is equilateral, isosceles or scalene triangle.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor:** File name: main.c. The code checks if three sides (side1, side2, side3) are equal to determine if it's an equilateral, isosceles, or scalene triangle.

```
1 #include <stdio.h>
2
3 int main() {
4     int side1, side2, side3;
5
6     printf("AADARSH CHAUDHARY\nEnter the lengths of all three sides of the triangle: ");
7     scanf("%d %d %d", &side1, &side2, &side3);
8
9     if (side1 == side2 && side2 == side3) {
10         printf("The triangle is an equilateral triangle!\n");
11     } else if (side1 == side2 || side2 == side3 || side1 == side3) {
12         printf("The triangle is an isosceles triangle!\n");
13     } else {
14         printf("The triangle is a scalene triangle!\n");
15     }
16
17     return 0;
18 }
```

- Output Window:** Shows the program's execution. The user enters "20 40 60" and the program outputs "The triangle is a scalene triangle!".

```
AADARSH CHAUDHARY
Enter the lengths of all three sides of the triangle: 20 40 60
The triangle is a scalene triangle!

...Program finished with exit code 0
Press ENTER to exit console.
```

19. Write a C program to find all roots of a quadratic equation.

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify, Download.
- Language:** C
- Code Editor:** File name: main.c. The code calculates the discriminant and finds two real roots or one repeated root if discriminant is zero, or two complex roots if discriminant is negative.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     float a, b, c;
6     float discriminant, root1, root2, realPart, imaginaryPart;
7
8     printf("AADARSH CHAUDHARY\nEnter the coefficients (a, b, c) of the quadratic equation: ");
9     scanf("%f %f %f", &a, &b, &c);
10
11    discriminant = b * b - 4 * a * c;
12
13    if (discriminant > 0) {
14        root1 = (-b + sqrt(discriminant)) / (2 * a);
15        root2 = (-b - sqrt(discriminant)) / (2 * a);
16        printf("The quadratic equation has two distinct real roots: %.2f and %.2f\n", root1, root2);
17    } else if (discriminant == 0) {
18        root1 = root2 = -b / (2 * a);
19        printf("The quadratic equation has a repeated real root: %.2f\n", root1);
20    } else {
21        realPart = -b / (2 * a);
22        imaginaryPart = sqrt(-discriminant) / (2 * a);
23        printf("The quadratic equation has two complex roots: %.2f + %.2fi and %.2f - %.2fi\n",
24               realPart, imaginaryPart, realPart, imaginaryPart);
25    }
26
27    return 0;
28 }
```

- Output Window:** Shows the program's execution. The user enters "2 2 2" and the program outputs "The quadratic equation has two complex roots: -0.50 + 0.87i and -0.50 - 0.87i".

```
AADARSH CHAUDHARY
Enter the coefficients (a, b, c) of the quadratic equation: 2 2 2
The quadratic equation has two complex roots: -0.50 + 0.87i and -0.50 - 0.87i
```

20. Write a C program to calculate profit or loss.

The screenshot shows a C IDE interface with a code editor and a terminal window. The code editor displays a file named 'main.c' containing a C program. The terminal window below it shows the execution of the program and its output.

```
main.c
1 #include <stdio.h>
2
3 int main() {
4     float costPrice, sellingPrice, profit, loss;
5
6     printf("AADARSH CHAUDHARY\nEnter the cost price: ");
7     scanf("%f", &costPrice);
8
9     printf("Now enter the selling price: ");
10    scanf("%f", &sellingPrice);
11
12    if (sellingPrice > costPrice) {
13        profit = sellingPrice - costPrice;
14        printf("Congratulations! You made a profit of %.2f\n", profit);
15    } else if (sellingPrice < costPrice) {
16        loss = costPrice - sellingPrice;
17        printf("Oops! You incurred a loss of %.2f\n", loss);
18    } else {
19        printf("It's a break-even situation! You neither made a profit nor incurred a loss.\n");
20    }
21
22    return 0;
23 }
```

input

```
AADARSH CHAUDHARY
Enter the cost price: 50
Now enter the selling price: 100
Congratulations! You made a profit of 50.00
```

21. Write a C program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer.

Calculate percentage and grade according to following:

Percentage $\geq 90\%$: Grade A Percentage $\geq 80\%$: Grade B

Percentage $\geq 70\%$: Grade C Percentage $\geq 60\%$: Grade D

Percentage $\geq 40\%$: Grade E Percentage $< 40\%$: Grade F

The screenshot shows a code editor window with a dark theme. At the top, there are several buttons: Run, Debug, Stop, Share, Save, and Beautify. The file tab shows "main.c". The code itself is a C program that prompts the user for five subject marks and calculates the total percentage and grade based on the given criteria. The output window at the bottom shows sample run data where the user inputs marks for Physics, Chemistry, Biology, Mathematics, and Computer, followed by the calculated percentage and grade.

```
1 #include<stdio.h>
2
3 int main() {
4     float physics, chemistry, biology, mathematics, computer;
5     float totalMarks, percentage;
6
7     printf("AADARSH CHAUDHARY\n");
8     printf("Enter the marks of Physics: ");
9     scanf("%f", &physics);
10
11    printf("Enter the marks of Chemistry: ");
12    scanf("%f", &chemistry);
13
14    printf("Enter the marks of Biology: ");
15    scanf("%f", &biology);
16
17    printf("Enter the marks of Mathematics: ");
18    scanf("%f", &mathematics);
19
20    printf("Enter the marks of Computer: ");
21    scanf("%f", &computer);
22
23    totalMarks = physics + chemistry + biology + mathematics + computer;
24    percentage = (totalMarks / 500) * 100;
25
26    printf("Percentage: %.2f\n", percentage);
27
28    if (percentage >= 90) {
29        printf("Grade: A\n");
30    }
31    else if (percentage >= 80) {
32        printf("Grade: B\n");
33    }
34    else if (percentage >= 70) {
35        printf("Grade: C\n");
36    }
37    else if (percentage >= 60) {
38        printf("Grade: D\n");
39    }
40    else if (percentage >= 40) {
41        printf("Grade: E\n");
42    }
43    else {
44        printf("Grade: F\n");
45    }
46
47    return 0;
48 }
```

AADARSH CHAUDHARY
Enter the marks of Physics: 96
Enter the marks of Chemistry: 92
Enter the marks of Biology: 93
Enter the marks of Mathematics: 90
Enter the marks of Computer: 95
Percentage: 93.20
Grade: A

22. Write a C program to input basic salary of an employee and calculate its Gross salary according to following: Basic Salary <= 10000 : HRA = 20%, DA = 80% Basic Salary <= 20000 : HRA = 25%, DA = 90% Basic Salary > 20000 : HRA = 30%, DA = 95%

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save), Run, Debug, Stop, Share, Save, and Beautify.
- Code Editor:** The file is named "main.c". The code implements a switch statement to calculate HRA and DA based on the basic salary. It then calculates the gross salary as the sum of basic salary, HRA, and DA.

```
1 #include<stdio.h>
2
3 int main() {
4     float basicSalary, grossSalary, hra, da;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter the basic salary of the employee: ");
8     scanf("%f", &basicSalary);
9
10    if (basicSalary <= 10000) {
11        hra = 0.2 * basicSalary;
12        da = 0.8 * basicSalary;
13    }
14    else if (basicSalary <= 20000) {
15        hra = 0.25 * basicSalary;
16        da = 0.9 * basicSalary;
17    }
18    else {
19        hra = 0.3 * basicSalary;
20        da = 0.95 * basicSalary;
21    }
22
23    grossSalary = basicSalary + hra + da;
24
25    printf("Gross Salary: %.2f\n", grossSalary);
26
27    return 0;
28 }
```

- Output Window:** Displays the program's output. It starts with the string "AADARSH CHAUDHARY", followed by the prompt "Enter the basic salary of the employee: ", and then the result "Gross Salary: 112500.00".

23. Write a C program to input electricity unit charges and calculate total electricity bill according to the given condition: For first 50 units Rs. 0.50/unit For next 100 units Rs. 0.75/unit For next 100 units Rs. 1.20/unit For unit above 250 Rs. 1.50/unit An additional surcharge of 20% is added to the bill

The screenshot shows a C IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save), run (Run, Debug, Stop, Share, Save, Beautify).
- Code Editor:** File named "main.c" containing the C program code.
- Code Content:**

```
1 #include <stdio.h>
2
3 int main() {
4     float units, bill;
5
6     printf("AADARSH CHAUDHARY\n");
7     printf("Enter the number of units consumed: ");
8     scanf("%f", &units);
9
10    if(units <= 50) {
11        bill = units * 0.50;
12    }
13    else if(units <= 150) {
14        bill = 50 * 0.50 + (units - 50) * 0.75;
15    }
16    else if(units <= 250) {
17        bill = 50 * 0.50 + 100 * 0.75 + (units - 150) * 1.20;
18    }
19    else {
20        bill = 50 * 0.50 + 100 * 0.75 + 100 * 1.20 + (units - 250) * 1.50;
21    }
22
23    bill += bill * 0.20; // Adding 20% surcharge
24
25    printf("Total electricity bill: Rs. %.2f\n", bill);
26
27    return 0;
28 }
```
- Output Window:** Displays the program's output:

```
AADARSH CHAUDHARY
Enter the number of units consumed: 1000
Total electricity bill: Rs. 1614.00

...Program finished with exit code 0
Press ENTER to exit console.
```

24. Write a C program to convert specified days into years, weeks and days. #include int main() { int days, years, weeks; days = 1329;

The screenshot shows a C code editor interface with the following details:

- Toolbar:** Includes icons for Save, Run, Debug, Stop, Share, Save, Beautify, and Download.
- Code Area:** The file is named "main.c". The code calculates the conversion of days into years, weeks, and remaining days. It includes a header file inclusion, variable declarations, user input for days, and a printf statement to output the results. The code is numbered from 1 to 19.
- Output Area:** Labeled "input" at the top right. It displays the program's output: "AADARSH CHAUDHARY", "Enter the number of days: 800", "800 days is equal to 2 years, 10 weeks, and 0 days.", followed by the standard completion message "...Program finished with exit code 0 Press ENTER to exit console."