LAB FILE

INTRODUCTION TO C PROGRAMMING

Batch : 2023 – 2027

BCA (Hons) AI & DS

Submitted by : Submitted to:

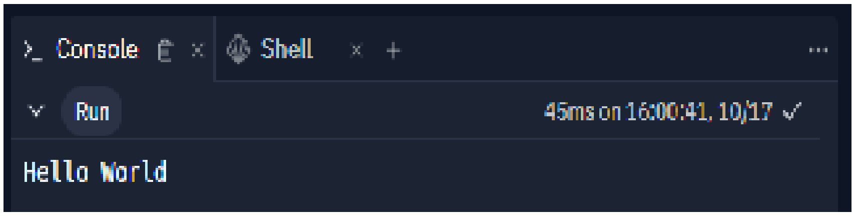
Gyanesh Jaiswal Mr. Rishi Kumar

Student ID: Assistant Prof.CSIT

2136012150   GEU

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

```
 #include <stdio.h>


int main(void) {
/*if digit - 1
  if not a digit - 0 */
  char ch;
  printf("enter the character ");
  scanf("%c",ch);
  printf("%d",(ch==0)&&(ch<=9));


  return 0;
}
```

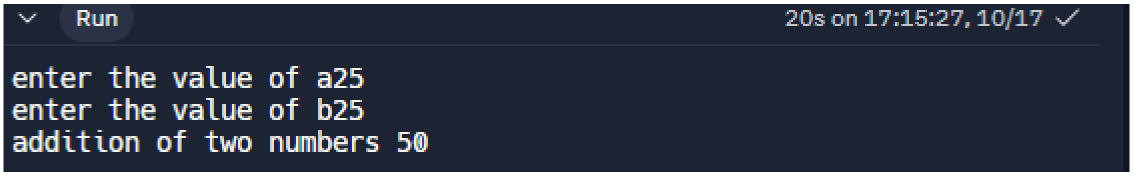


```
Hello World
```

2. WAP to add two Numbers.

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

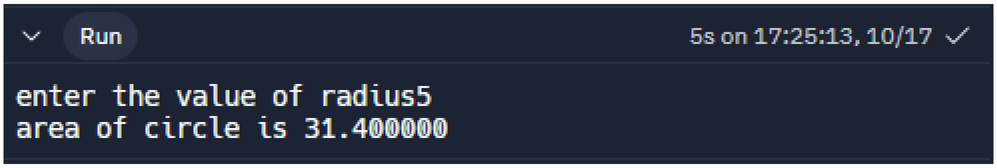
```
//WAP to add two numbers
int main(void)
{
  int add,a,b;
  printf("enter the value of a");
  scanf("%d",&a);
  printf("enter the value of b");
  scanf("%d",&b);
  add=a+b;
  printf("addition of two numbers %d",add);
  return 0;
}
```



```
enter the value of a25
enter the value of b25
addition of two numbers 50
```

3.WAP to find area of Circle.

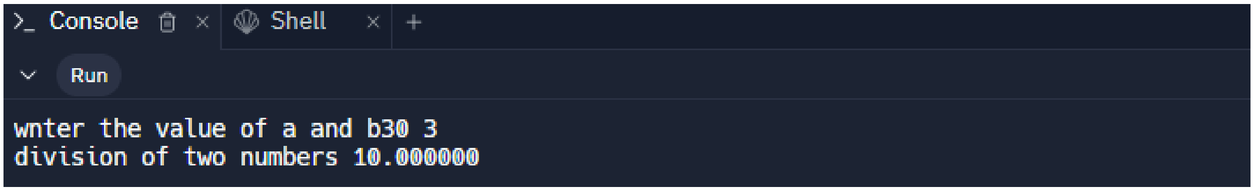
```
#include <stdio.h>
//WAP to find area of circle
int main(void)
{
  float area,radius;
  printf("enter the value of radius");
  scanf("%f",&radius);
  area=2*3.14*radius;
  printf("area of circle is %f",area);
  return 0;

}
```



```
enter the value of radius5
area of circle is 31.400000
```

4.WAP to divide two Numbers.

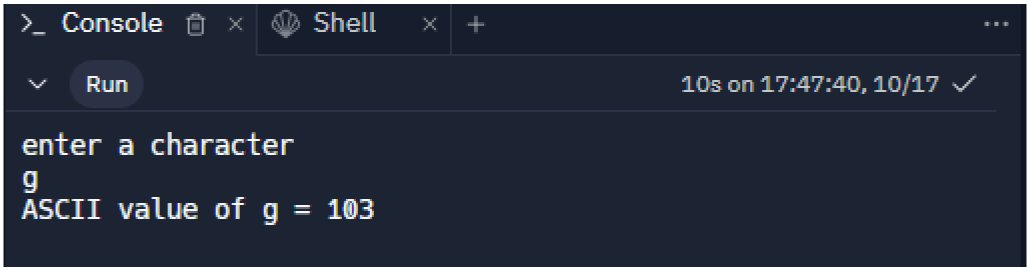
```
#include <stdio.h>
//WAP to find divide two numbers
Int main ()
{
float divide ,a,b;
printf(“enter the value of a and b”);
scanf(“%f%f”, &a,&b);
divide = a/b;
printf(“division of two numbers are :%f”,divide);
return 0;
}
```



```
wnter the value of a and b30 3
division of two numbers 10.000000
```

5.WAP to print ASCII value.

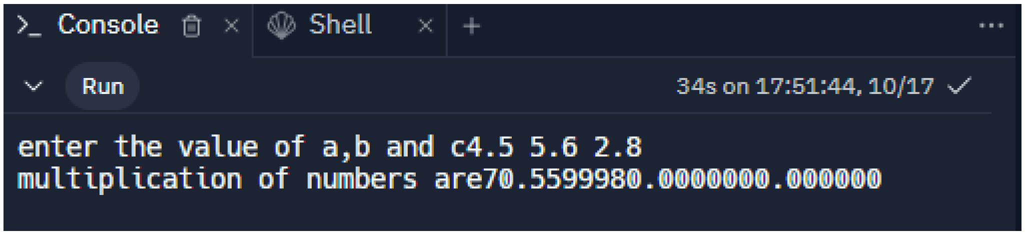
```
#include <stdio.h>
// Q 5. WAP to print ASCII value
int main(void) {
char c;
  printf("enter a character\n");
  scanf("%c", &c);
  printf("ASCII value of %c = %d",c,c);
  return 0;
}
```



```
enter a character
g
ASCII value of g = 103
```

6.WAP to multiply floating point Number.

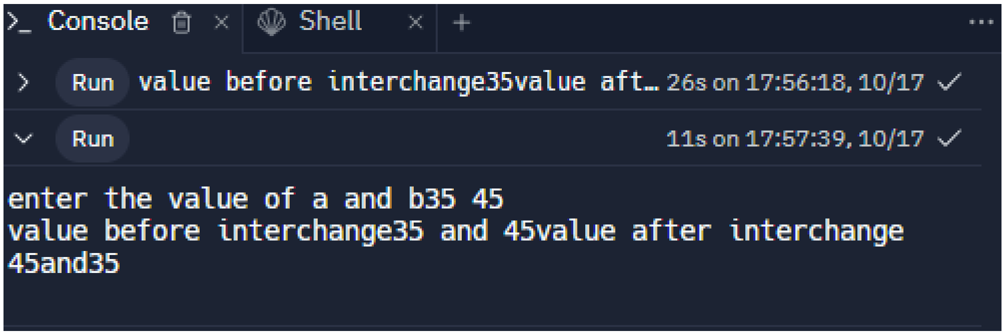
```
#include <stdio.h>
// WAP to multiply floating point numbers
int main(void) {
  float multiply,a,b,c;
  printf("enter the value of a,b and c");
  scanf(" %f%f%f",&a ,&b, &c);
  printf("multiplication of numbers 
are%f%f%f",multiply=a*b*c);
  return 0;
}
```



```
enter the value of a,b and c4.5 5.6 2.8
multiplication of numbers are70.5599980.0000000.000000
```

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

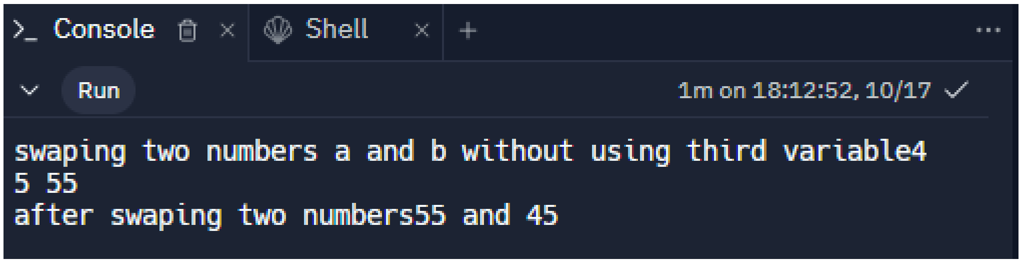
```
#include <stdio.h>
//WAP to SWAP two variable number by using third variable
int main(void) {
int a,b,c;
  printf("enter the value of a and b");
scanf("%d%d",&a,&b);
  printf("value before interchange%d and %d",a,b);
 c=a;
 a=b;
 b=c;
  printf("value after interchange %dand%d",a,b);
return 0;
}
```



```
Run value before interchange35value aft… 26s on 17:56:18, 10/17
Run 11s on 17:57:39, 10/17
enter the value of a and b35 45
value before interchange35 and 45value after interchange
45and35
```

8. WAP to SWAP two variable numbers without using third variable.

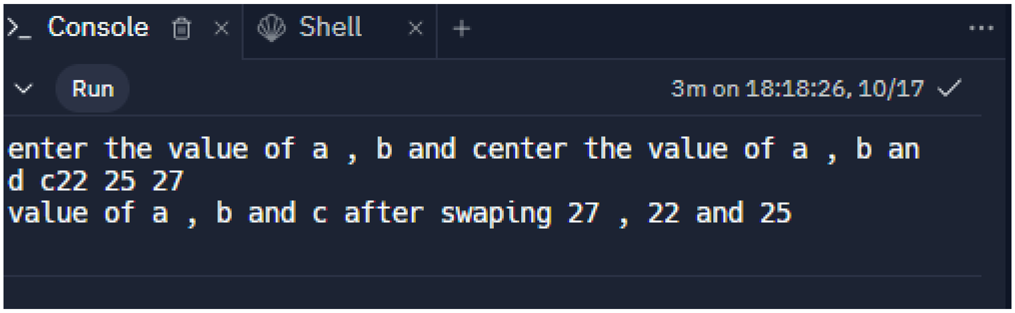
```
#include <stdio.h>
//WAP to SWAP two variable number without using third
variable
int main(void) {
  int a,b;
  printf("swaping two numbers a and b without using third
variable");
  scanf("%d%d",&a,&b);
  a=a+b;
  b=a-b;
  a=a-b;
  printf("after swaping two numbers%d and %d",a,b);
  return 0;
}
```



```
swaping two numbers a and b without using third variable4
5 55
after swaping two numbers55 and 45
```

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

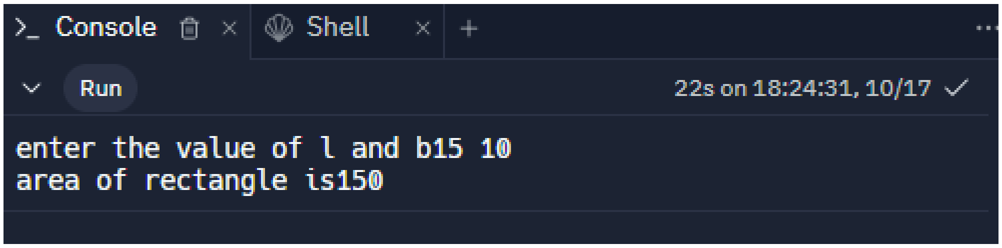
```
#include <stdio.h>
//WAP to SWAP 3 variable number without using 3 variable
int main(void) {
  int a,b,c;
  printf("enter the value of a , b and c");
  scanf("%d%d%d",&a,&b,&c);
  a=a+b+c;
  b=a-b-c;
  c=a-b-c;
  a=a-b-c;
  printf("value of a , b and c after swaping %d , %d and
%d",a,b,c);
  return 0;
}
```



```
enter the value of a , b and center the value of a , b an
d c22 25 27
value of a , b and c after swaping 27 , 22 and 25
```

10. WAP to find the area of rectangle.

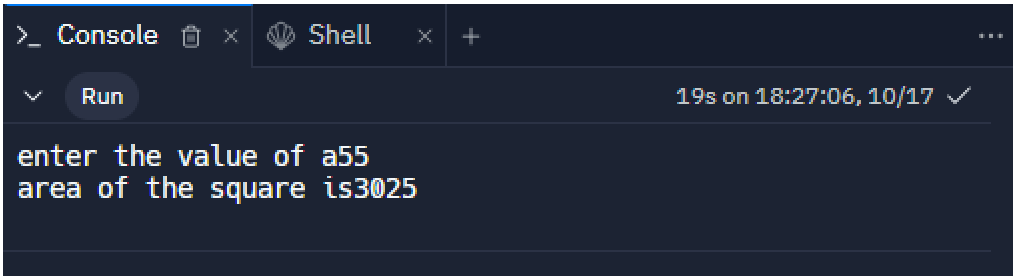
```
#include <stdio.h>
//WAP to find area of rectangle
int main(void) {
int l,b,area;
  printf("enter the value of l and b");
  scanf("%d%d",&l,&b);

  area = l*b;

  printf("area of rectangle is%d",area);
  return 0;
}
```



```
enter the value of l and b15 10
area of rectangle is150
```

11.WAP to find area of square.

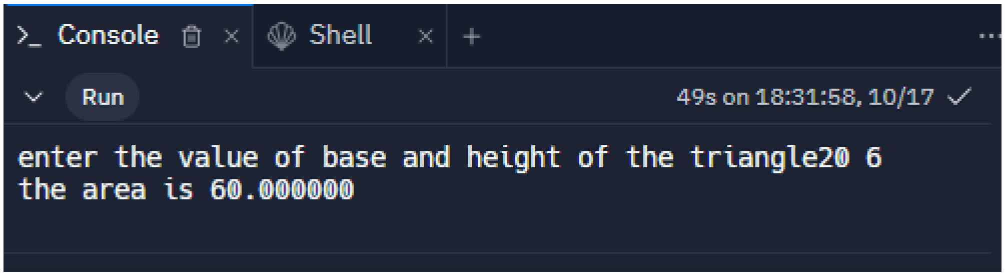
```
#include <stdio.h>
//WAP to find area of square
int main(void) {
  int area,a;
  printf("enter the value of a");
  scanf("%d",&a);
  area=a*a;
  printf("area of the square is%d",area);
  return 0;
}
```



```
enter the value of a55
area of the square is3025
```

12.WAP to find area of right angle triangle, isosceles triangle , any triangle with third side.

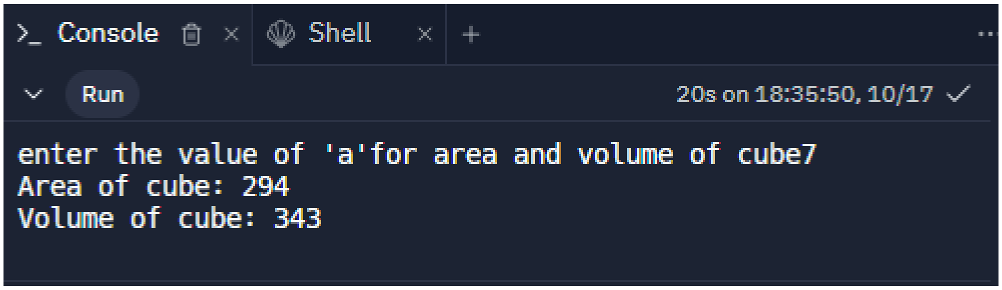
```
#include <stdio.h>

int main() {
  //WAP to find area of right angel triangle, isosceles
triangle,any triangle with 3 sides
  float area,base,height;
  printf("enter the value of base and height of the triangle");
  scanf("%f%f",&base,&height);
  area = (0.5)*base*height;
printf("the area is %f", area);
  return 0;
}
```



```
enter the value of base and height of the triangle20 6
the area is 60.000000
```

13. WAP to find area and volume of cube.

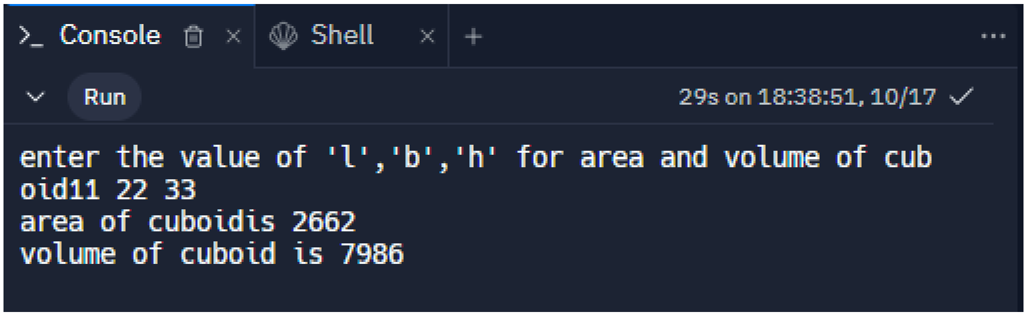
```
#include <stdio.h>
// WAP to find area and volume of cube
int main(void) {
 int area,volume,a;
  printf("enter the value of 'a'for area and volume of cube");
  scanf("%d",&a);
  area=6*a*a;
volume=a*a*a;
  printf("Area of cube: %d\n",area);
  printf("Volume of cube: %d\n",volume);
  return 0;
}
```



```
enter the value of 'a'for area and volume of cube7
Area of cube: 294
Volume of cube: 343
```

14.WAP to find area and volume of cuboid.

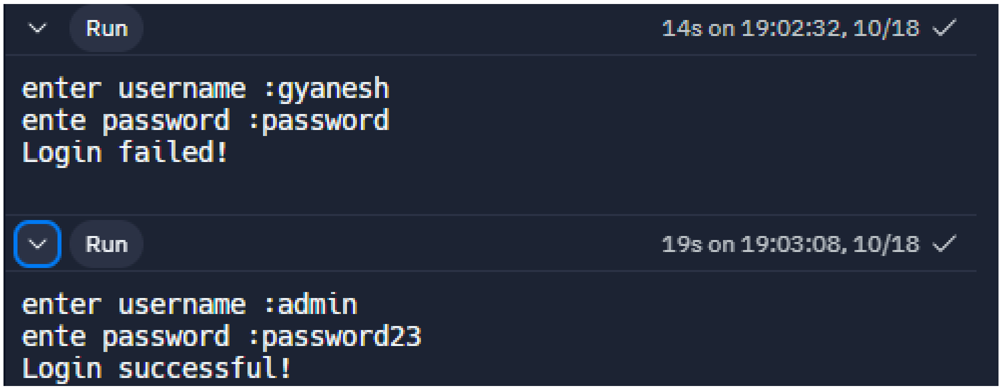
```
#include <stdio.h>
//WAP to find area and volume of cuboid
int main(void)
 {
  int  area, volume, l, b, h;
  printf("enter the value of 'l', 'b', 'h' for area and volume of
cuboid");
  scanf("%d%d%d",&l,&b,&h);
  area=2*(l*b+b*h+h*l);
  volume=l*b*h;
  printf("area of cuboidis %d\n",area);
printf("volume of cuboid is %d",volume);
  return 0;
}
```



```
enter the value of 'l','b','h' for area and volume of cub
oid11 22 33
area of cuboidis 2662
volume of cuboid is 7986
```

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

```
#include <stdio.h>
#include <string.h>
//WAP to validate the username and password entered by the user is correct or not using the predefined usernam and password.
int main(void) {
  char username[] = "admin";
  char password[] = "password23";
  char inputusername[50];
  char inputpassword[50];
  printf("enter username :");
  scanf("%s",inputusername);
  printf("ente password :");
  scanf("%s",inputpassword);
  if (strcmp(inputusername, username) == 0 && strcmp(inputpassword , password) == 0){
    printf("Login successful!\n");
  }else{
    printf("Login failed!\n");
  }
  return 0;
}
```

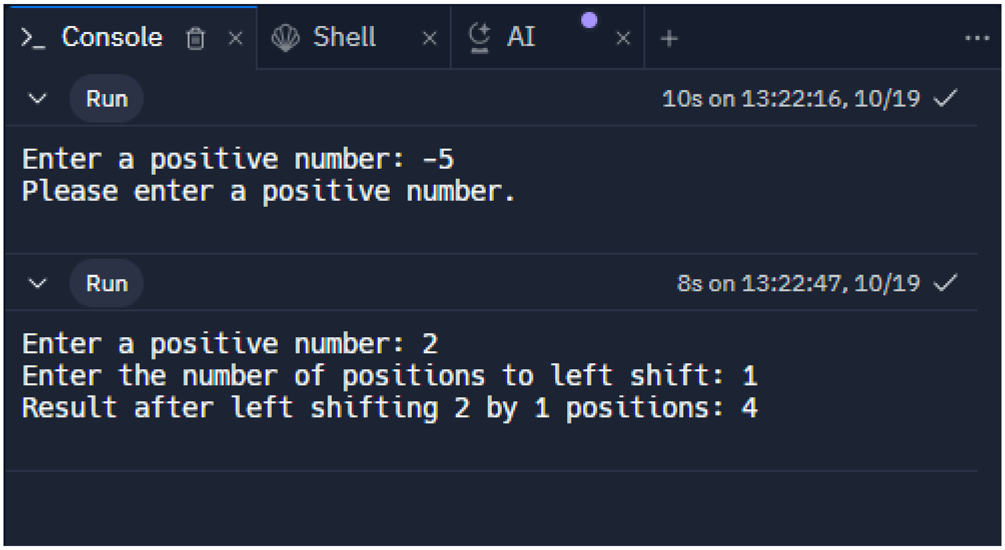


```
enter username :gyanesh
ente password :password
Login failed!
```

```
enter username :admin
ente password :password23
Login successful!
```

17**. WAP to input the positive number from the user to perform the left shit operator.**

```
#include <stdio.h>
// WAP to input the positive number from the user to perform the left shit operator.
int main() {
    int num, shift;

    // Take input for the positive number
    printf("Enter a positive number: ");
    scanf("%d", &num);
    // Check if the number is positive
    if (num < 0) {
        printf("Please enter a positive number.\n");
    } else {
        // Take input for the shift count
        printf("Enter the number of positions to left shift: ");
        scanf("%d", &shift);

        // Perform left shift operation
        int result = num << shift;

        // Display the result
        printf("Result after left shifting %d by %d positions: 
%d\n", num, shift, result);
    }

    return 0;
}
```

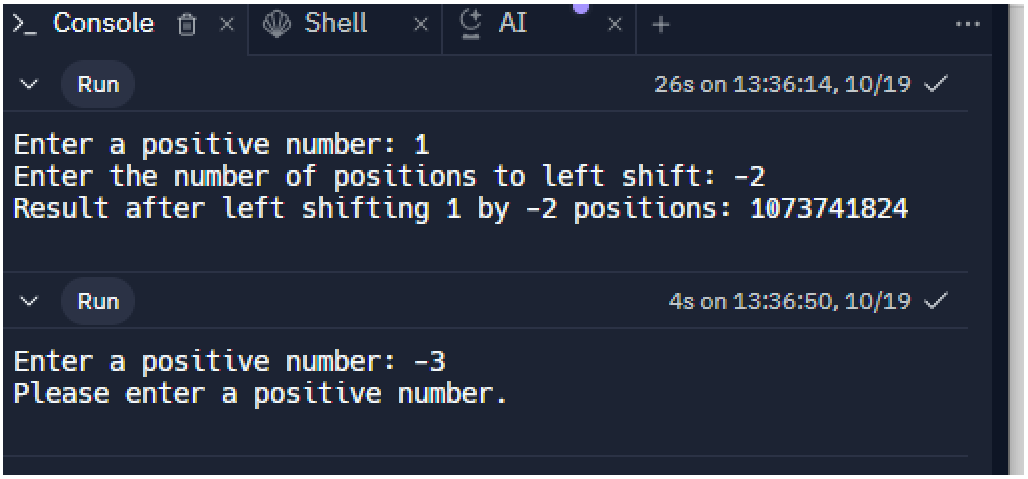


```
Enter a positive number: -5
Please enter a positive number.
```

```
Enter a positive number: 2
Enter the number of positions to left shift: 1
Result after left shifting 2 by 1 positions: 4
```

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

```
#include <stdio.h>
//. WAP to input the positive number from the user to perform the Right shit operator.
int main() {
    int num, shift;

    // Take input for the positive number
    printf("Enter a positive number: ");
    scanf("%d", &num);

    // Check if the number is positive
    if (num < 0) {
        printf("Please enter a positive number.\n");
    } else {
        // Take input for the shift count
        printf("Enter the number of positions to left shift: ");
        scanf("%d", &shift);

        // Perform left shift operation
        int result = num << shift;

        // Display the result
        printf("Result after left shifting %d by %d positions: %d\n", num, shift, result);
    }
}
```

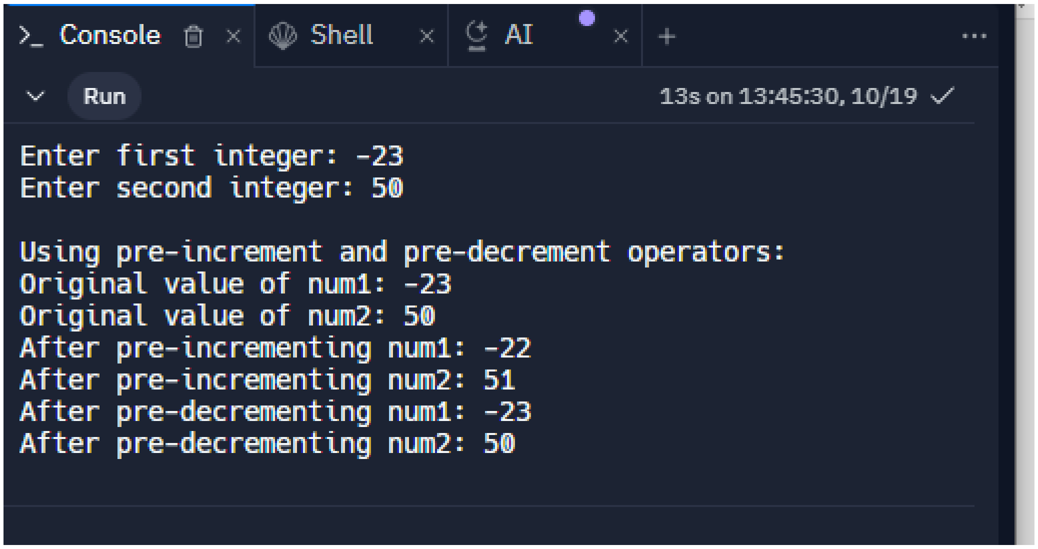


```
Enter a positive number: 1
Enter the number of positions to left shift: -2
Result after left shifting 1 by -2 positions: 1073741824
```

```
Enter a positive number: -3
Please enter a positive number.
```

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

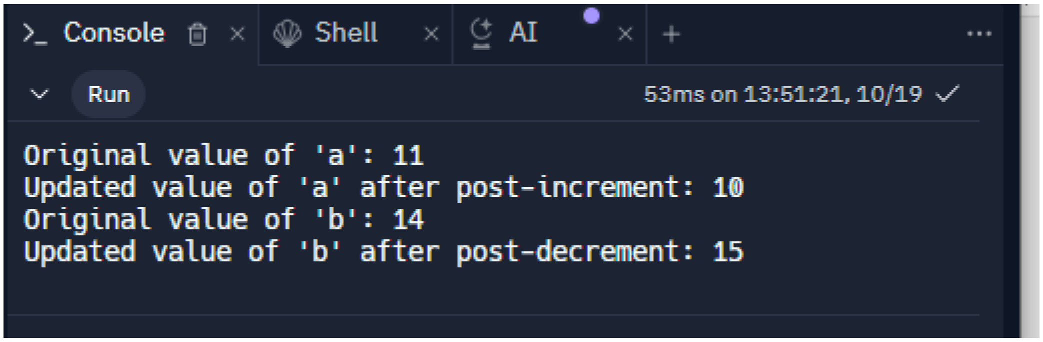
```
#include <stdio.h>

int main() {
    int num1, num2;

    // Input two integers
    printf("Enter first integer: ");
    scanf("%d", &num1);
    printf("Enter second integer: ");
    scanf("%d", &num2);

    // Pre-increment and Pre-decrement operations
    printf("\nUsing pre-increment and pre-decrement
operators:\n");
    printf("Original value of num1: %d\n", num1);
    printf("Original value of num2: %d\n", num2);

    // Pre-increment
    printf("After pre-incrementing num1: %d\n",
++num1);
    printf("After pre-incrementing num2: %d\n",
++num2);

    // Pre-decrement
    printf("After pre-decrementing num1: %d\n", --
num1);
    printf("After pre-decrementing num2: %d\n", --
num2);

    return 0;
}
```



```
Enter first integer: -23
Enter second integer: 50

Using pre-increment and pre-decrement operators:
Original value of num1: -23
Original value of num2: 50
After pre-incrementing num1: -22
After pre-incrementing num2: 51
After pre-decrementing num1: -23
After pre-decrementing num2: 50
```

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

```
#include <stdio.h>

int main() {
    int a = 10,b = 15;
    // Post-increment and post-decrement operations
    int postIncA = a++;
    int postDecB = b--;

    // Print original values and updated values
    printf("Original value of 'a': %d\n", a);
    printf("Updated value of 'a' after post-increment:
%d\n", postIncA);
    printf("Original value of 'b': %d\n", b);
    printf("Updated value of 'b' after post-decrement:
%d\n", postDecB);

    return 0;
}
```

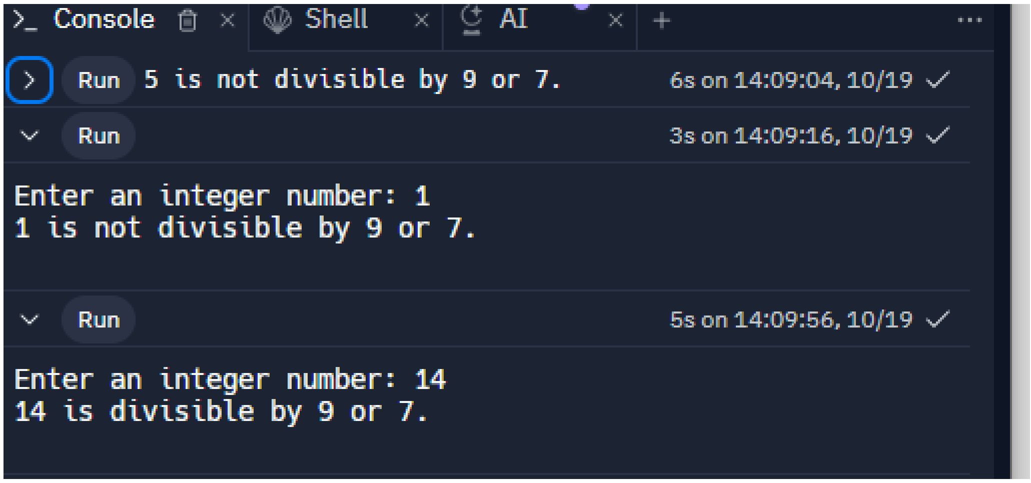


```
Original value of 'a': 11
Updated value of 'a' after post-increment: 10
Original value of 'b': 14
Updated value of 'b' after post-decrement: 15
```

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

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

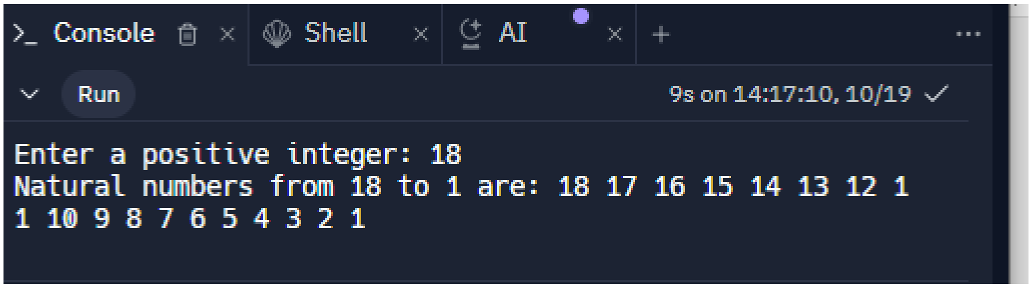
```
int main() {
    int number;

    printf("Enter an integer number: ");
    scanf("%d", &number);



    if (number % 9 == 0 || number % 7 == 0) {
        printf("%d is divisible by 9 or 7.\n", number);
    } else {
        printf("%d is not divisible by 9 or 7.\n", number);
    }

    return 0;
}
```



```
5 is not divisible by 9 or 7.        6s on 14:09:04, 10/19
                                     3s on 14:09:16, 10/19
Enter an integer number: 1
1 is not divisible by 9 or 7.
                                     5s on 14:09:56, 10/19
Enter an integer number: 14
14 is divisible by 9 or 7.
```

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

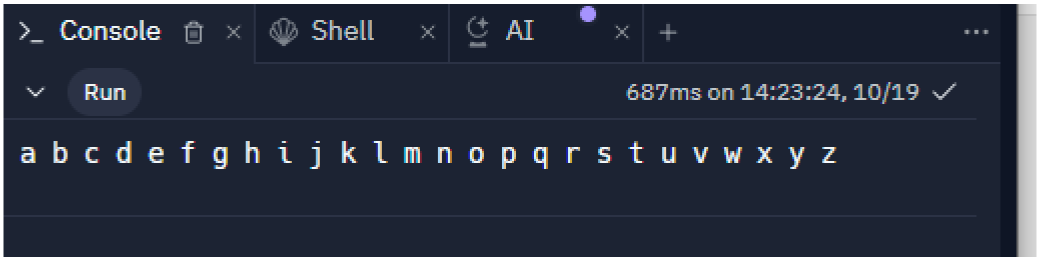
```
#include <stdio.h>

int main() {
    int n, i;

    printf("Enter a positive integer: ");
    scanf("%d", &n);

    printf("Natural numbers from %d to 1 are: ", n);
    for (i = n; i >= 1; i--) {
        printf("%d ", i);
    }

    return 0;
}
```



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

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

```
#include <stdio.h>

int main() {
    char alphabet;

    // Print all alphabets from 'a' to 'z'
    for (alphabet = 'a'; alphabet <= 'z'; ++alphabet) {
        printf("%c ", alphabet);
    }

    // Print a new line at the end
    printf("\n");

    return 0;
}
```

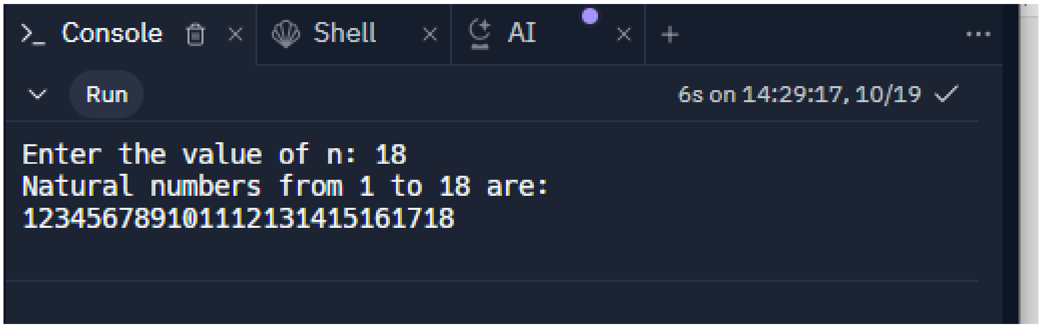


```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

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

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

```
int main() {
    int n, i;

    // Ask the user to enter the value of n
    printf("Enter the value of n: ");
    scanf("%d", &n);

    // Iterate from 1 to n and print each number
    printf("Natural numbers from 1 to %d are: \n", n);
    for (i = 1; i <= n; i++) {
        printf("%d\n", i);
    }

    return 0;
}
```

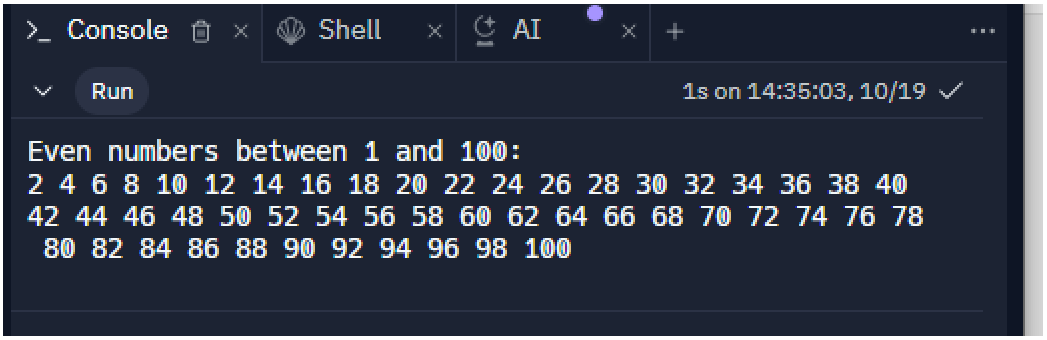


```
Enter the value of n: 18
Natural numbers from 1 to 18 are:
123456789101112131415161718
```

27. **Write a program to print all even number between 1 to 100.**

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

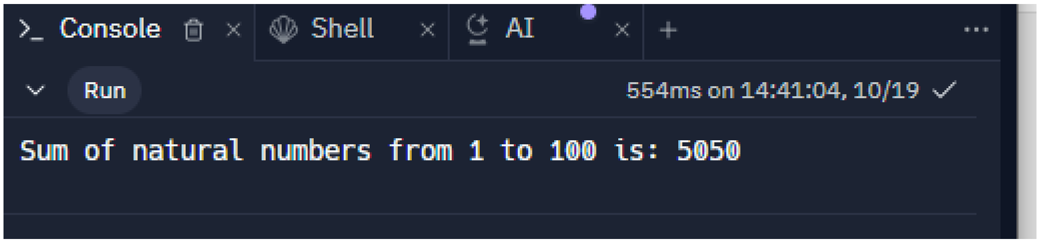
```
int main() {
    int i;

    printf("Even numbers between 1 and 100:\n");

    // Loop through numbers from 1 to 100
    for (i = 1; i <= 100; i++) {
        // Check if the number is even
        if (i % 2 == 0) {
            // If even, print the number
            printf("%d ", i);
        }
    }
        return 0;
}
```



```
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
```

28. Write a C program to find sum of all natural number between 1 to 100.

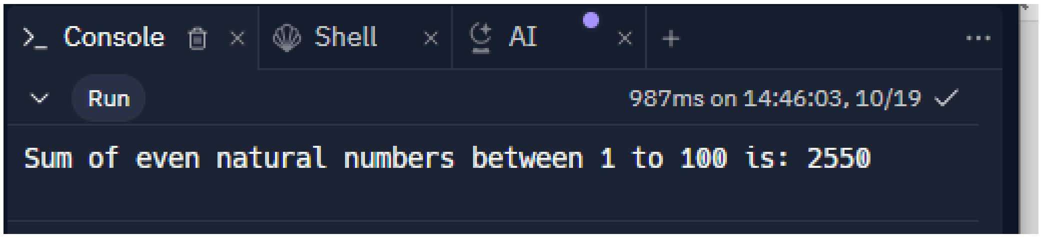
```
#include <stdio.h>

int main() {
    int i, sum = 0;

    // Loop through numbers from 1 to 100 and calculate the sum
    for(i = 1; i <= 100; i++) {
        sum += i; // Add current number to the sum
    }

    // Print the result
    printf("Sum of natural numbers from 1 to 100 is: %d\n", sum);

    return 0;
}
```



```
Sum of natural numbers from 1 to 100 is: 5050
```

29. Write a C program to find sum of all even natural number between 1 to 100.

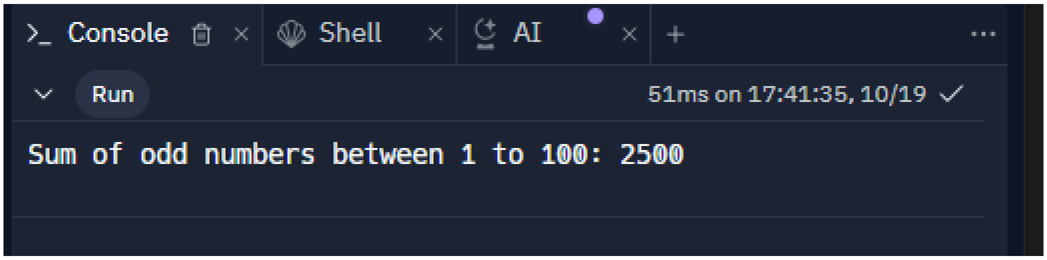
```
#include <stdio.h>

int main() {
    int sum = 0;
    int i;

    // Loop through numbers from 1 to 100
    for(i = 1; i <= 100; i++) {
        // Check if the number is even
        if(i % 2 == 0) {
            // If even, add it to the sum
            sum += i;
        }
    }

    // Print the sum of even numbers
    printf("Sum of even natural numbers between 1 to 100
is: %d\n", sum);

    return 0;
}
```



```
Sum of even natural numbers between 1 to 100 is: 2550
```

30. Write a program to find sum of all odd numbers between 1 to 100.

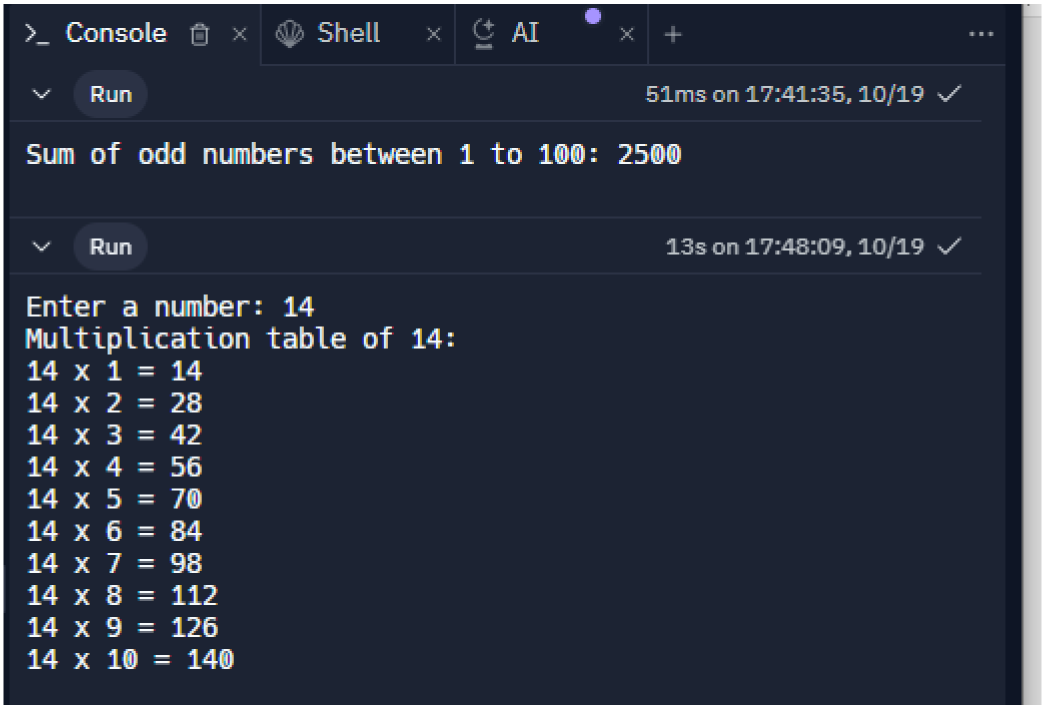
```
#include <stdio.h>

int main() {
    int sum = 0 ;
    int i;

    // Loop through numbers from 1 to 100
    for (i = 1; i <= 100; i++) {
        // Check if the number is odd
        if (i % 2 != 0) {
            // Add the odd number to the sum
            sum += i;
        }
    }

    // Print the sum of odd numbers
    printf("Sum of odd numbers between 1 to 100: %d\n", sum);

    return 0;
}
```



```
Sum of odd numbers between 1 to 100: 2500
```

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

```
#include <stdio.h>

int main() {
    int num, i;

    // Ask the user to input a number
    printf("Enter a number: ");
    scanf("%d", &num);

    // Print multiplication table from 1 to 10
    printf("Multiplication table of %d:\n", num);
    for (i = 1; i <= 10; i++) {
        printf("%d x %d = %d\n", num, i, num * i);
    }

    return 0;
}
```



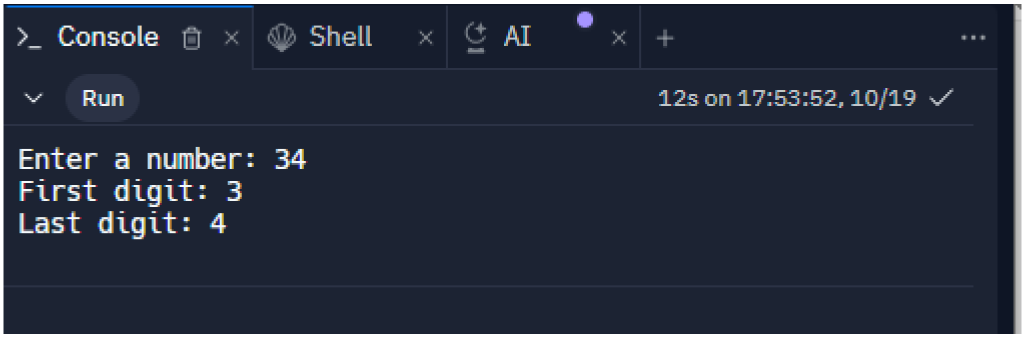
```
Sum of odd numbers between 1 to 100: 2500
```

```
Enter a number: 14
Multiplication table of 14:
14 x 1 = 14
14 x 2 = 28
14 x 3 = 42
14 x 4 = 56
14 x 5 = 70
14 x 6 = 84
14 x 7 = 98
14 x 8 = 112
14 x 9 = 126
14 x 10 = 140
```

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

```
#include <stdio.h>

int main() {
   int number, firstDigit, lastDigit;

   // Input number from user
   printf("Enter a number: ");
   scanf("%d", &number);

   // Extracting the last digit
```

```
    lastDigit = number % 10;

    // Finding the first digit
    while (number >= 10) {
        number /= 10;
    }
    firstDigit = number;

    // Output the first and last digit
    printf("First digit: %d\n", firstDigit);
    printf("Last digit: %d\n", lastDigit);

    return 0;
}
```

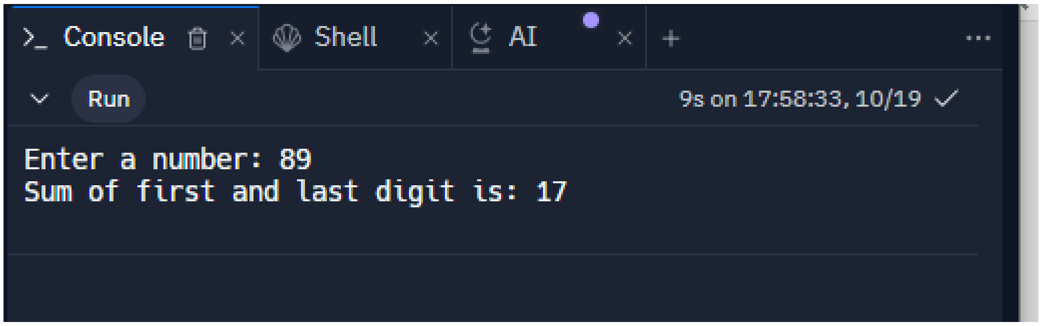


```
Enter a number: 34
First digit: 3
Last digit: 4
```

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

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

```
int main() {
    int number, firstDigit, lastDigit, sum;

    // Read the number from the user
    printf("Enter a number: ");
    scanf("%d", &number);

    // Find the last digit of the number
    lastDigit = number % 10;

    // Find the first digit of the number
    while (number >= 10) {
        number /= 10;
    }
    firstDigit = number;

    // Calculate the sum of the first and last digit
    sum = firstDigit + lastDigit;

    // Display the result
    printf("Sum of first and last digit is: %d\n", sum);

    return
    0;
}
```

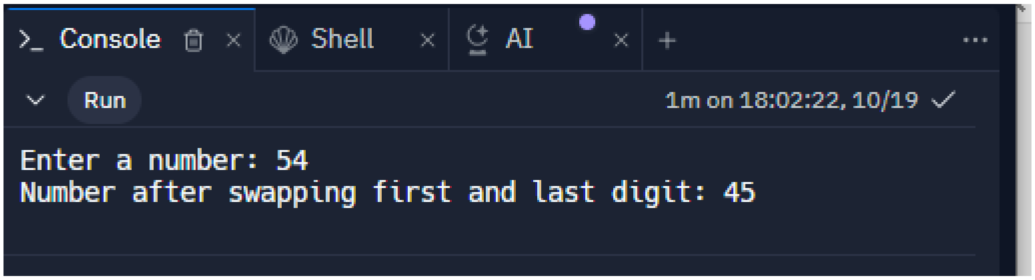


```
Enter a number: 89
Sum of first and last digit is: 17
```

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

```
#include <stdio.h>

int main() {
   int number, originalNumber, lastDigit, firstDigit,
swappedNumber = 0, multiplier = 1;

   // Input a number from the user
   printf("Enter a number: ");
   scanf("%d", &number);

   // Store the original number for later use
   originalNumber = number;

   // Find the last digit of the number
   lastDigit = number % 10;

   // Count the number of digits in the number
```

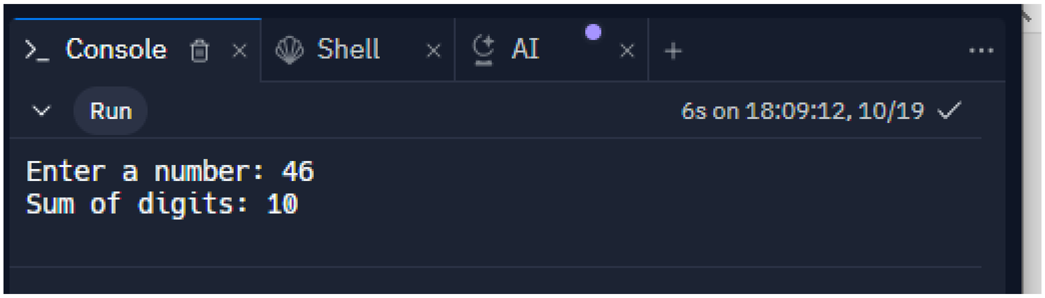
```
    while (number >= 10) {
        number /= 10;
        multiplier *= 10;
    }

    // Find the first digit of the original number
    firstDigit = number;

    // Swap the first and last digits
    swappedNumber = lastDigit * multiplier +
originalNumber % multiplier;
    swappedNumber -= lastDigit;
    swappedNumber += firstDigit;

    // Output the swapped number
    printf("Number after swapping first and last digit:
%d\n", swappedNumber);

    return 0;
}
```



```
Enter a number: 54
Number after swapping first and last digit: 45
```

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

```
#include <stdio.h>

int main() {
    int num, digit, sum = 0;

    // Input number from user
    printf("Enter a number: ");
    scanf("%d", &num);

    // Calculate sum of digits
    while (num > 0) {
        digit = num % 10; // Extract the last digit
        sum += digit;      // Add the digit to the sum
        num /= 10;        // Remove the last digit from the
number
    }

    // Print the sum of digits
    printf("Sum of digits: %d\n", sum);

    return 0;
}
```

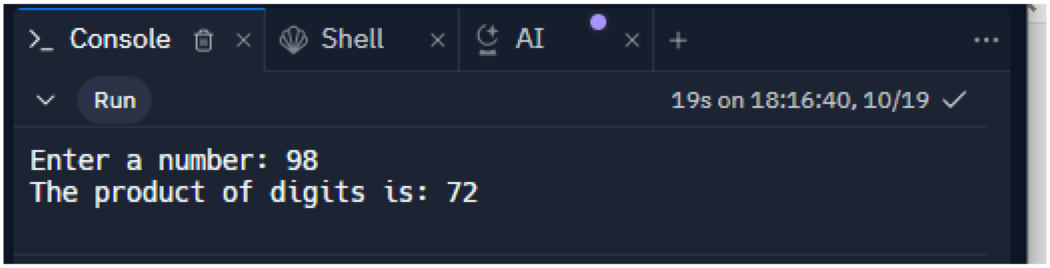


```
Enter a number: 46
Sum of digits: 10
```

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

```
#include <stdio.h>

int main() {
   int number, digit, product = 1;

   printf("Enter a number: ");
   scanf("%d", &number);

   while (number != 0) {

      digit = number % 10;

      product *= digit;

      number /= 10;
   }
   printf("The product of digits is: %d\n", product);
```

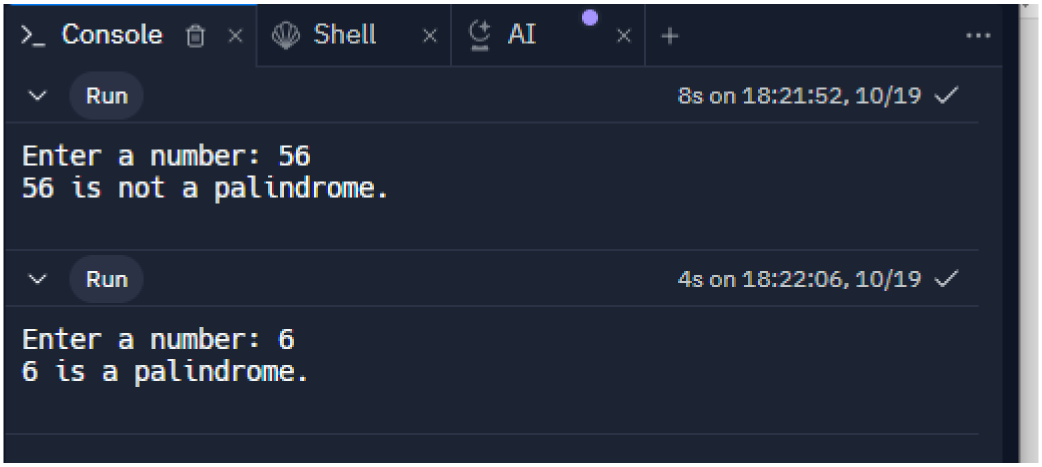
```
    return 0;
}
```



```
Enter a number: 98
The product of digits is: 72
```

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

```
#include <stdio.h>

int isPalindrome(int num) {
    int reversedNum = 0, originalNum = num;

    while (num > 0) {
        int digit = num % 10;
        reversedNum = reversedNum * 10 + digit;
        num /= 10;
    }

    if (originalNum == reversedNum) {
        return 1; // It is a palindrome
    } else {
        return 0; // It is not a palindrome
    }
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    if (isPalindrome(num)) {
        printf("%d is a palindrome.\n", num);
    } else {
        printf("%d is not a palindrome.\n", num);
    }

    return 0;
}
```

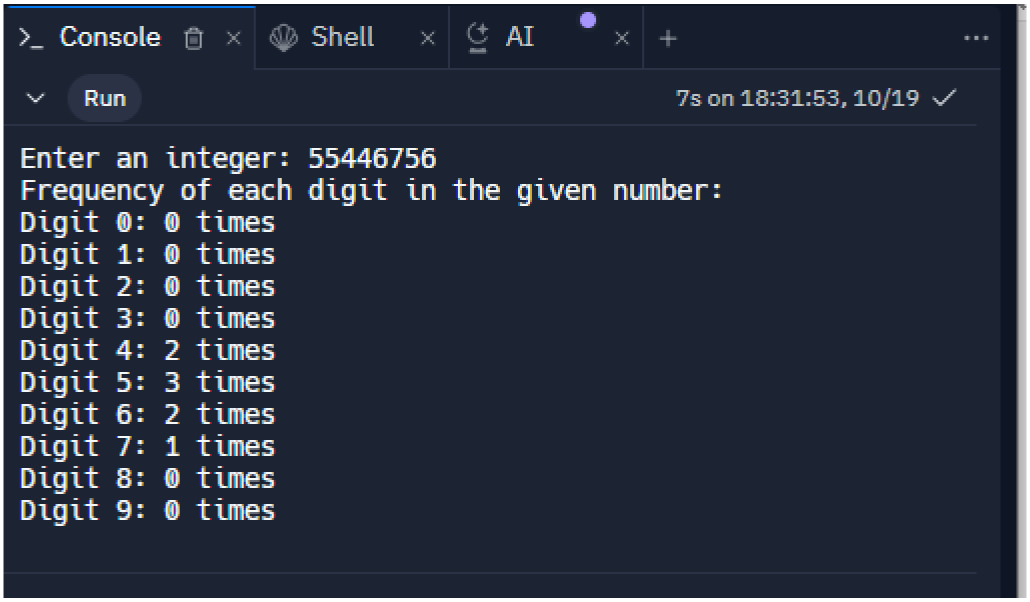


```
Enter a number: 56
56 is not a palindrome.
```

```
Enter a number: 6
6 is a palindrome.
```

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

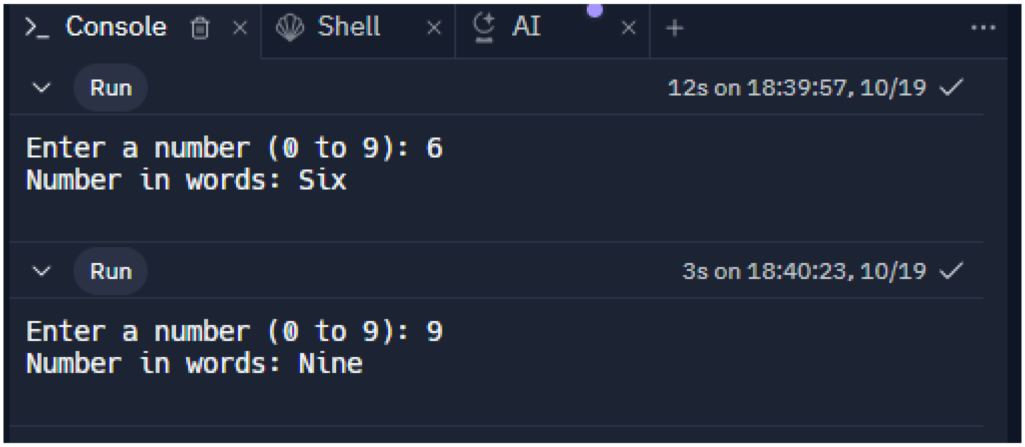
```
#include <stdio.h>

int main() {
    int num, digit, i;
    int frequency[10] = {0};
    digits (0 to 9)
    printf("Enter an integer: ");
    scanf("%d", &num);

    // Calculate the frequency of each digit
    while (num != 0) {
        digit = num % 10;
        frequency[digit]++;
        num = num / 10;
}
    printf("Frequency of each digit in the given
number:\n");
    for (i = 0; i < 10; i++) {
        printf("Digit %d: %d times\n", i, frequency[i]);
    }

    return 0;
}
```



```
Enter an integer: 55446756
Frequency of each digit in the given number:
Digit 0: 0 times
Digit 1: 0 times
Digit 2: 0 times
Digit 3: 0 times
Digit 4: 2 times
Digit 5: 3 times
Digit 6: 2 times
Digit 7: 1 times
Digit 8: 0 times
Digit 9: 0 times
```

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

```
#include <stdio.h>

void printNumberInWords(int num) {
   char *words[] = {"Zero", "One", "Two", "Three",
"Four", "Five", "Six", "Seven", "Eight", "Nine"};

   if (num < 0 || num > 9) {
      printf("Number out of range (0 to 9).\n");
   } else {
      printf("Number in words: %s\n", words[num]);
   }
}

int main() {
   int number;

   printf("Enter a number (0 to 9): ");
   scanf("%d", &number);

   printNumberInWords(number);

   return 0;
}
```



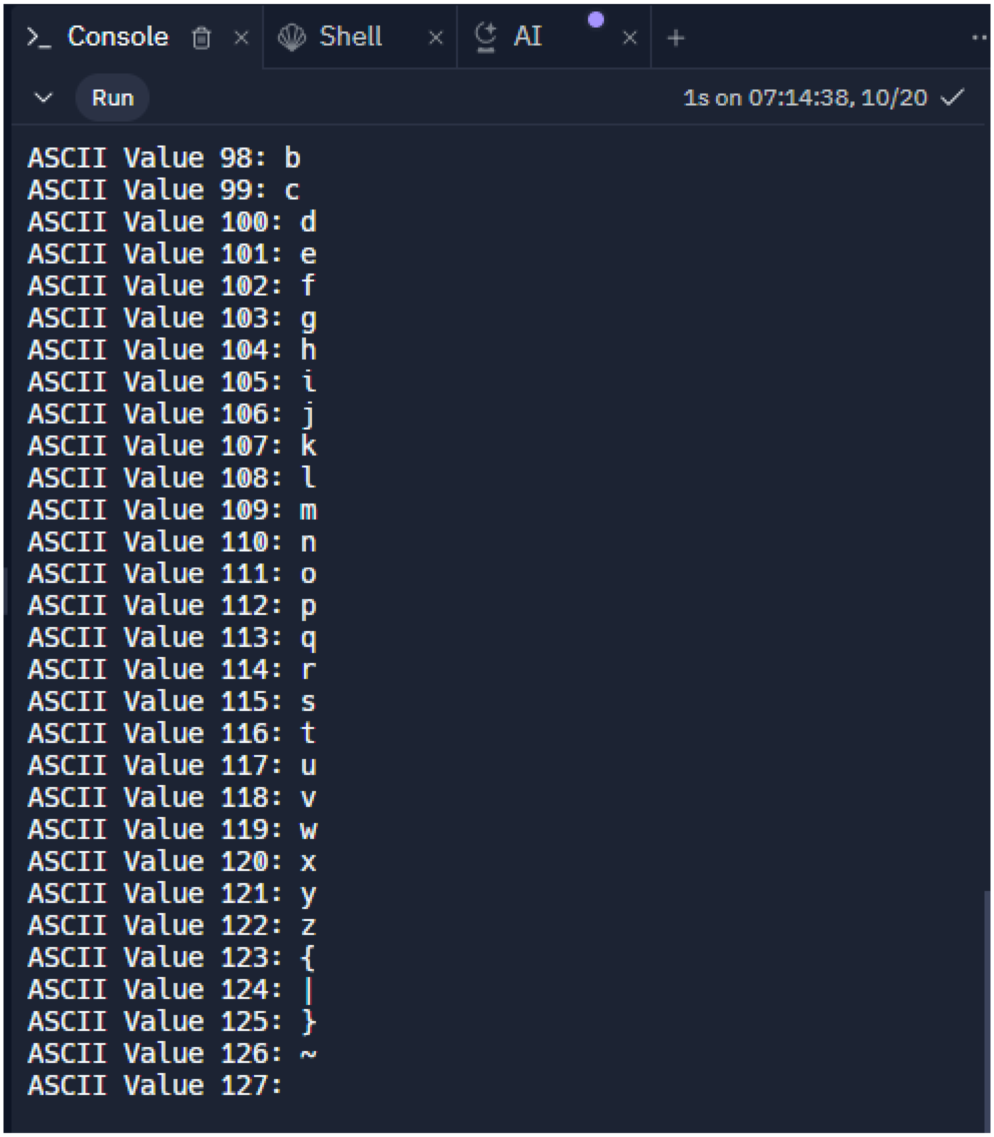
```
Enter a number (0 to 9): 6
Number in words: Six
```

```
Enter a number (0 to 9): 9
Number in words: Nine
```

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

```
#include <stdio.h>

int main() {
   printf("ASCII Characters and Their Values (0-127):\n");
```

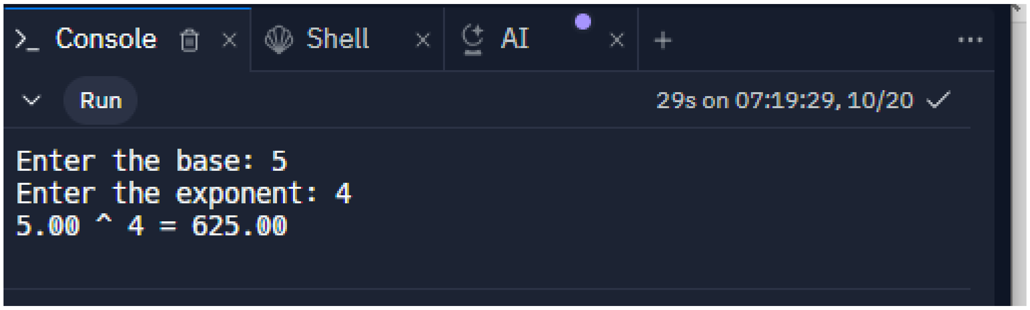
```
    for (int i = 0; i <= 127; i++) {
        printf("ASCII Value %d: %c\n", i, (char)i);
    }

    return 0;
}
```



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

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

```
#include <stdio.h>

double power(double base, int exponent) {
    double result = 1.0;
    for (int i = 0; i < exponent; i++) {
        result *= base;
    }

    return result;
}

int main() {
    double base, result;
    int exponent;
    printf("Enter the base: ");
    scanf("%lf", &base);
    printf("Enter the exponent: ");
    scanf("%d", &exponent);
    result = power(base, exponent);
    printf("%.2lf ^ %d = %.2lf\n", base, exponent, result);

    return 0;
}
```

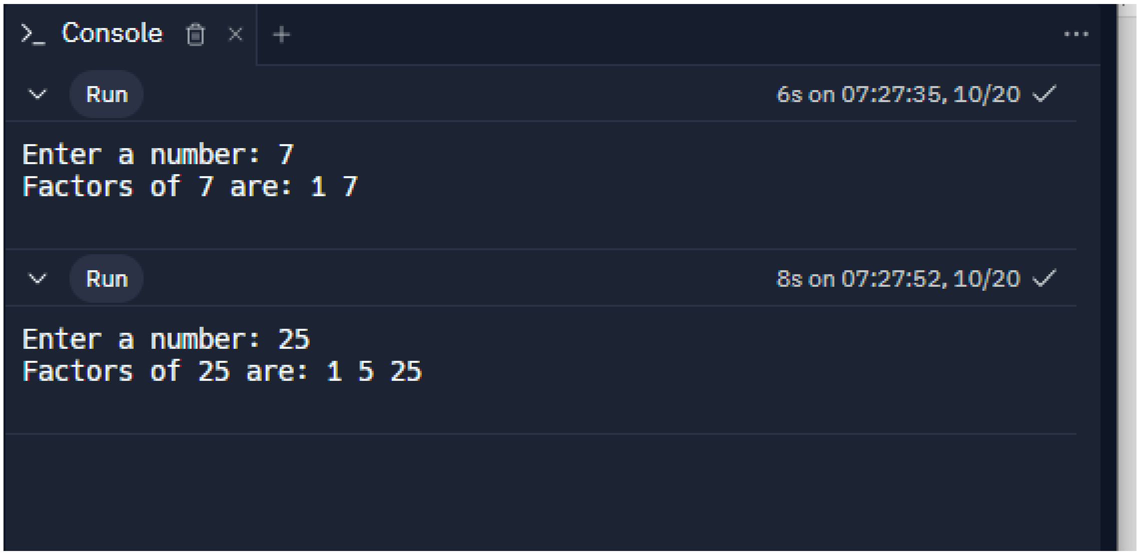


```
Enter the base: 5
Enter the exponent: 4
5.00 ^ 4 = 625.00
```

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

```
#include <stdio.h>

int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);

    printf("Factors of %d are: ", number);
    for (int i = 1; i <= number; i++) {
        if (number % i == 0) {
            printf("%d ", i);
        }
    }
    printf("\n");
```

```
  return 0;
}
```

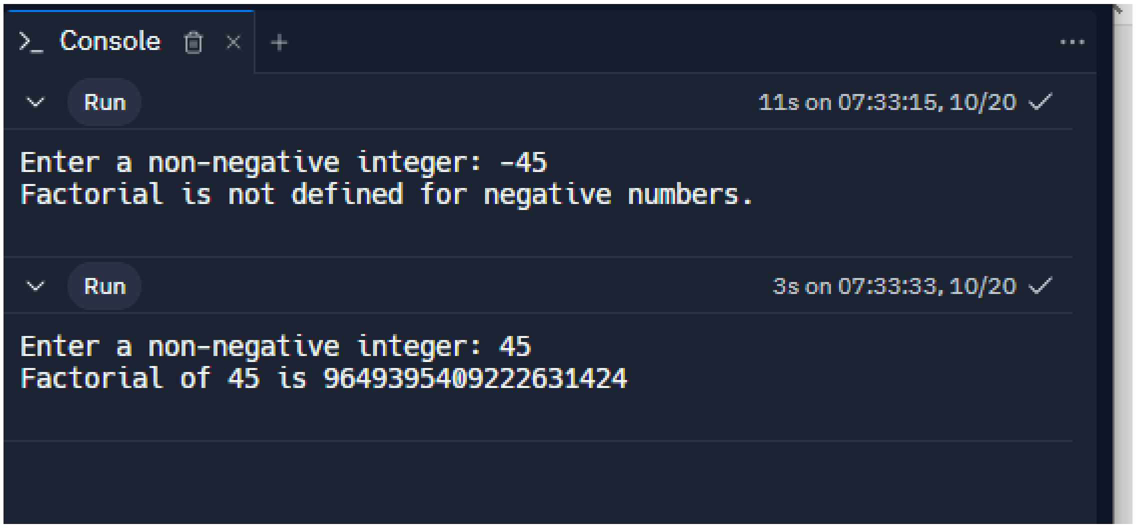


```
Enter a number: 7
Factors of 7 are: 1 7
```

```
Enter a number: 25
Factors of 25 are: 1 5 25
```

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

```
#include <stdio.h>

int main() {
  int n;
  unsigned long long factorial = 1;
  printf("Enter a non-negative integer: ");
  scanf("%d", &n);
  if (n < 0) {
    printf("Factorial is not defined for negative numbers.\n");
  } else {
    for (int i = 1; i <= n; i++) {
      factorial *= i;
    }
    printf("Factorial of %d is %llu\n", n, factorial);
  }
  return 0;
}
```

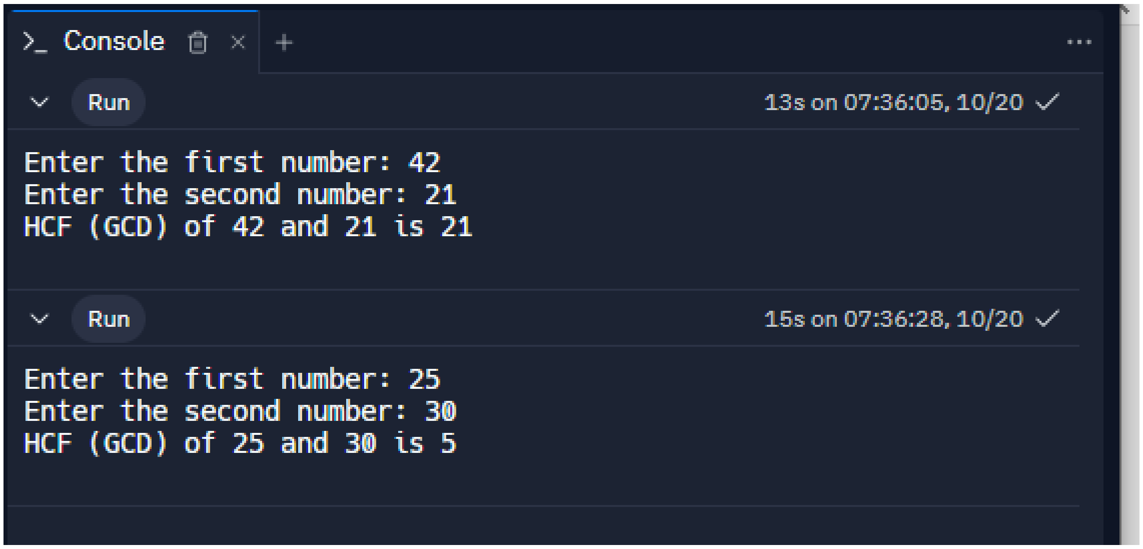


```
Enter a non-negative integer: -45
Factorial is not defined for negative numbers.
```

```
Enter a non-negative integer: 45
Factorial of 45 is 9649395409222631424
```

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

```
#include <stdio.h>

int findHCF(int a, int b) {
   while (a != b) {
      if (a > b) {
         a = a - b;
      } else {
         b = b - a;
      }
   }
   return a;
}

int main() {
   int num1, num2;
   printf("Enter the first number: ");
   scanf("%d", &num1);
   printf("Enter the second number: ");
   scanf("%d", &num2);
   int hcf = findHCF(num1, num2);
   printf("HCF (GCD) of %d and %d is %d\n", num1, num2, hcf);

   return 0;
}
```

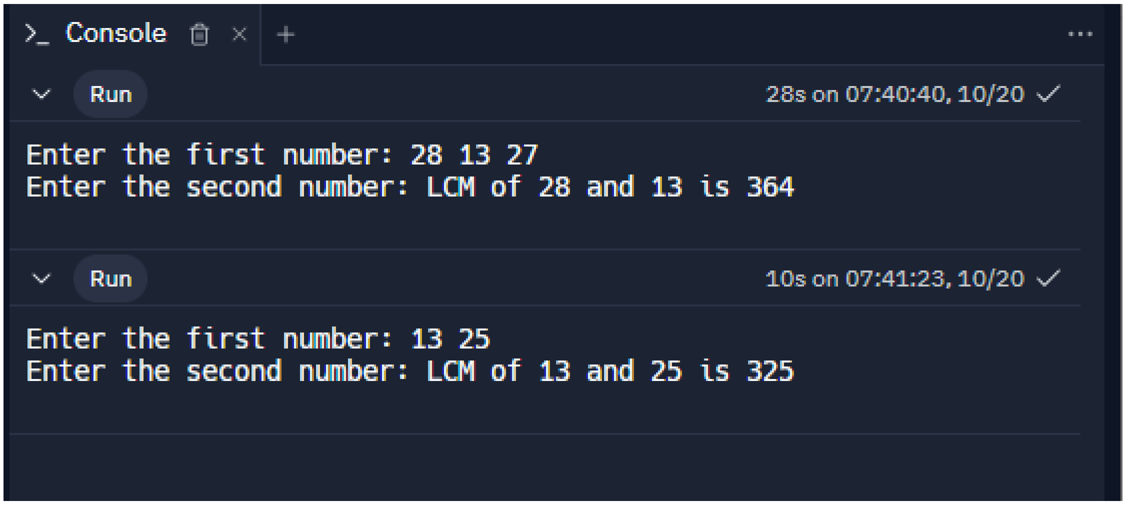


```
Enter the first number: 42
Enter the second number: 21
HCF (GCD) of 42 and 21 is 21
```

```
Enter the first number: 25
Enter the second number: 30
HCF (GCD) of 25 and 30 is 5
```

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

```
#include <stdio.h>

int findHCF(int a, int b) {
  while (a != b) {
    if (a > b) {
      a = a - b;
    } else {
      b = b - a;
    }
  }
  return a;
}

int findLCM(int a, int b) {
  int hcf = findHCF(a, b);
  int lcm = (a * b) / hcf;
  return lcm;
}

int main() {
  int num1, num2;
  printf("Enter the first number: ");
  scanf("%d", &num1);
  printf("Enter the second number: ");
  scanf("%d", &num2);
  int lcm = findLCM(num1, num2);
  printf("LCM of %d and %d is %d\n", num1, num2, lcm);

  return 0;  }
```

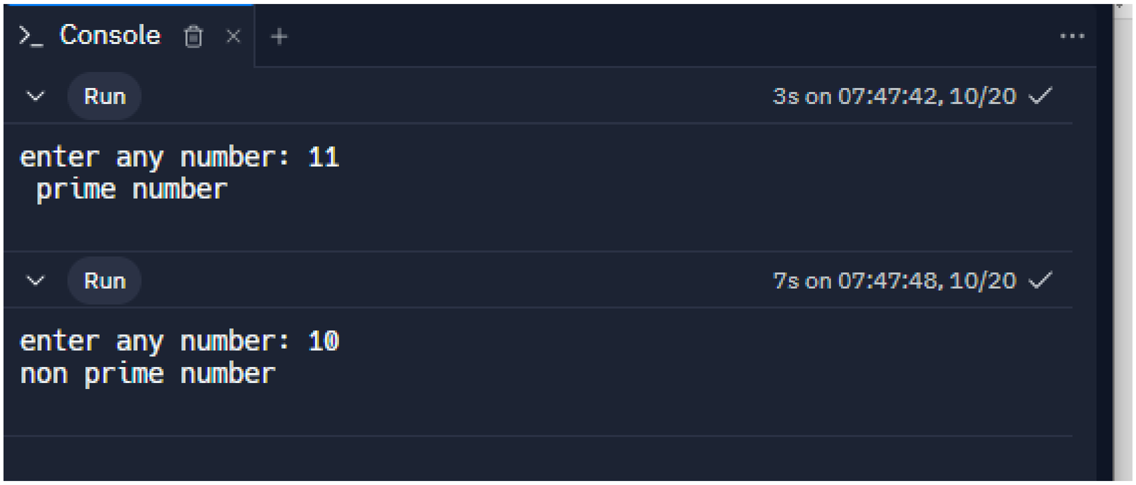


```
Enter the first number: 28 13 27
Enter the second number: LCM of 28 and 13 is 364
```

```
Enter the first number: 13 25
Enter the second number: LCM of 13 and 25 is 325
```

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

```
#include <stdio.h>

  int main(){

   int n,i,count=0;

  printf ("enter any number: ");

   scanf ("%d",&n);

   for(i=1;i<=n;i++){

       if (n%i==1){

           count++;

       }

   }if (count==2)

   printf ("non prime number");
   else

   printf (" prime number");

   return 0;

}
```



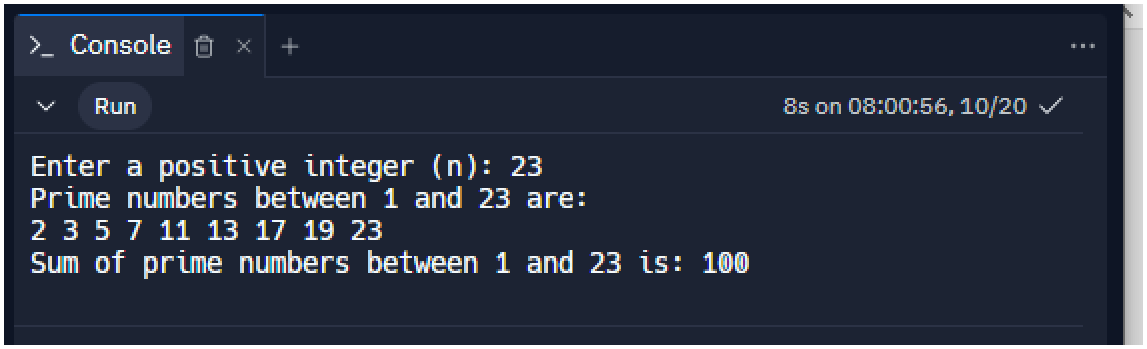
```
enter any number: 11
 prime number
```

```
enter any number: 10
non prime number
```

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

```
#include <stdio.h>

int Prime(int num) {
   if (num <= 1) {
      return 0;
   }
   for (int i = 2; i * i <= num; i++) {
      if (num % i == 0) {
        return 0;
      }
   }
   return 1;
}

int main() {
   int n, sum = 0;
```

```
    printf("Enter a positive integer (n): ");
    scanf("%d", &n);

    if (n <= 1) {
        printf("There are no prime numbers in the range 1 to %d\n", n);
    } else {
        printf("Prime numbers between 1 and %d are:\n", n);
        for (int i = 2; i <= n; i++) {
            if (isPrime(i)) {
                printf("%d ", i);
                sum += i;
            }
        }
        printf("\n");
        printf("Sum of prime numbers between 1 and %d is: %d\n", n, sum);
    }

    return 0;
}
```

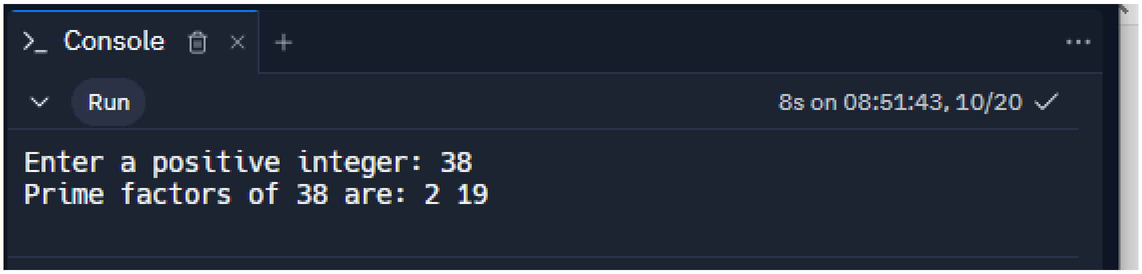


```
Enter a positive integer (n): 23
Prime numbers between 1 and 23 are:
2 3 5 7 11 13 17 19 23
Sum of prime numbers between 1 and 23 is: 100
```

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

```
#include <stdio.h>

int isPrime(int num) {
    if (num <= 1) {
        return 0;
    }
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) {
            return 0;
        }
    }
    return 1;
}
```

```
int main() {
   int number;
   printf("Enter a positive integer: ");
   scanf("%d", &number);

   printf("Prime factors of %d are: ", number);

   if (number <= 1) {
      printf("There are no prime factors for %d.\n", number);
   } else {
      for (int i = 2; i <= number; i++) {
         if (number % i == 0 && isPrime(i)) {
            while (number % i == 0) {
               printf("%d ", i);
               number = number / i;
            }
         }
      }
      printf("\n");
   }

   return 0;
}
```



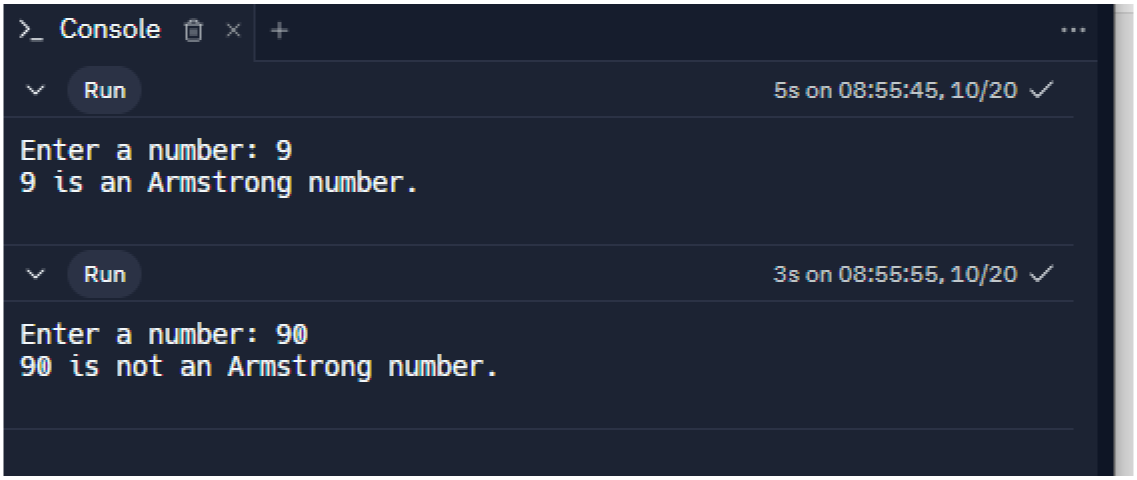
```
Enter a positive integer: 38
Prime factors of 38 are: 2 19
```

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

```
#include <stdio.h>

#include <math.h>

int isArmstrong(int num) {
```

```
int originalNum, remainder, result = 0, n = 0;

originalNum = num;

  while (originalNum != 0) {

    originalNum /= 10;

    ++n;

}

    originalNum = num;

    while (originalNum != 0) {

    remainder = originalNum % 10;

    result += pow(remainder, n);

   originalNum /= 10;


   }

   if (result == num)

    return 1;

   else

    return 0;
```

```
}

int main() {
    int num;
    printf("Enter a number: ");
   scanf("%d", &num);
    if (isArmstrong(num))
     printf("%d is an Armstrong number.\n", num);
   else
     printf("%d is not an Armstrong number.\n", num);
   return 0;


}
```



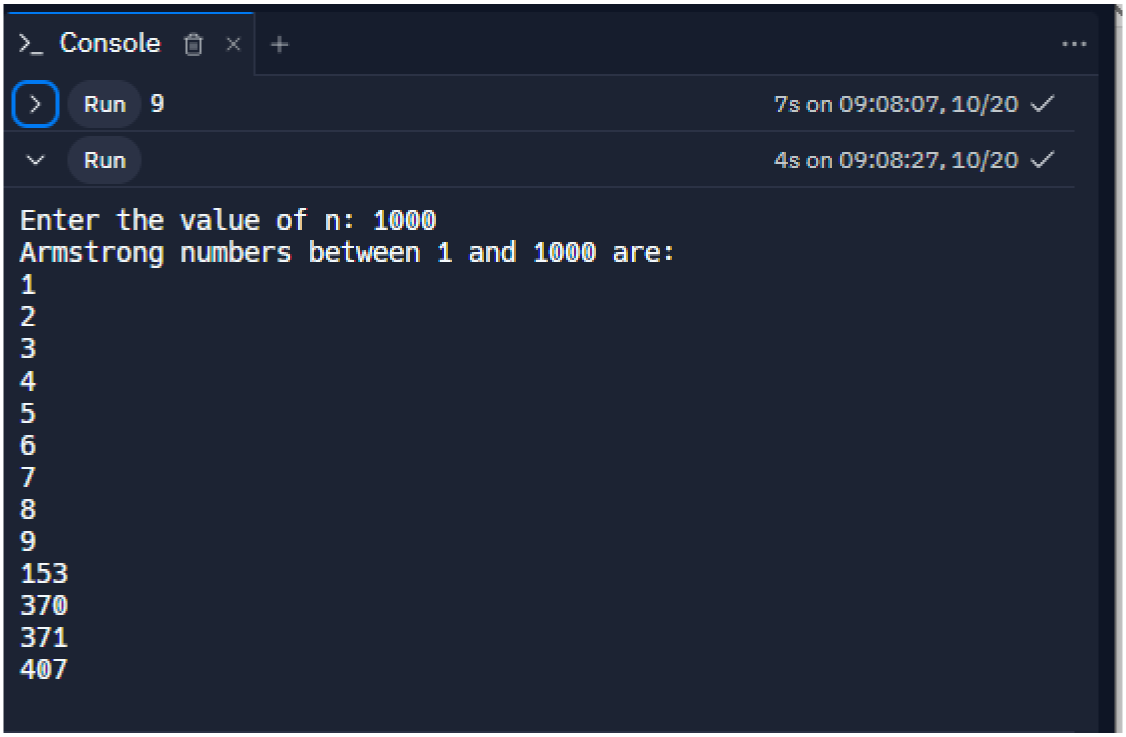
```
Enter a number: 9
9 is an Armstrong number.
```

```
Enter a number: 90
90 is not an Armstrong number.
```

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

```
#include <stdio.h>

#include <math.h>


int isArmstrong(int num) {


    int originalNum, remainder, result = 0, n = 0;


    originalNum = num;


    while (originalNum != 0) {

        originalNum /= 10;

        ++n;

    }


    originalNum = num;
    while (originalNum != 0) {

        remainder = originalNum % 10;
        result += pow(remainder, n);
        originalNum /= 10;
    }
    if (result == num)
        return 1;
    else
        return 0;

}

int main() {

    int n, i;
    printf("Enter the value of n: ");
    scanf("%d", &n);
```

```
 printf("Armstrong numbers between 1 and %d are:\n", n);
    for (i = 1; i <= n; i++) {

        if (isArmstrong(i)) {
            printf("%d\n", i);
        }
    }
    return 0;
}
```



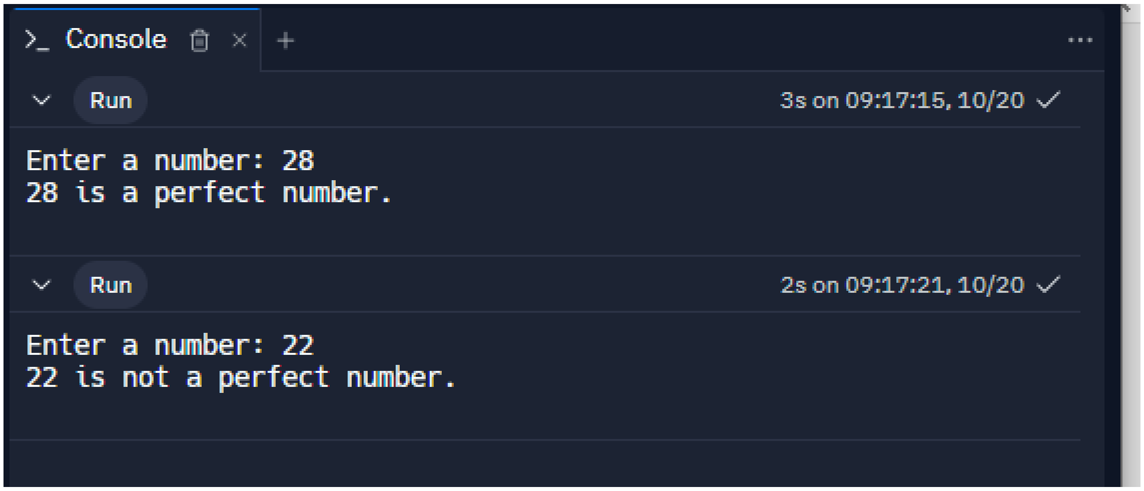
```
Enter the value of n: 1000
Armstrong numbers between 1 and 1000 are:
1
2
3
4
5
6
7
8
9
153
370
371
407
```

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

```c
#include <stdio.h>


int isPerfect(int num) {
    int sum = 0;

    for (int i = 1; i <= num / 2; i++) {
        if (num % i == 0) {
            sum += i;
        }
    }
    if (sum == num)
{
     return 1;
     } else
```

```
 {
        return 0;
    }
}
int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);
    if (isPerfect(num))
 {
    printf("%d is a perfect number.\n", num);
        } else
{
        printf("%d is not a perfect number.\n", num);
    }
    return 0;
```

```
}
```



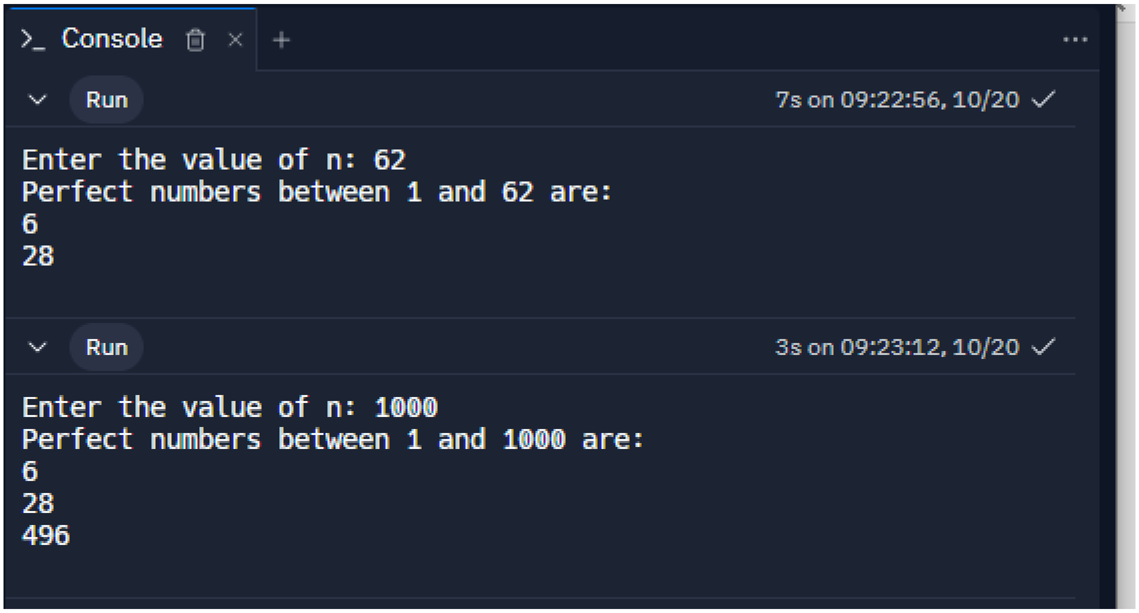
```
Enter a number: 28
28 is a perfect number.
```

```
Enter a number: 22
22 is not a perfect number.
```

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

```
#include <stdio.h>

int isPerfect(int num) {

    int sum = 0;



   for (int i = 1; i <= num / 2; i++) {



       if (num % i == 0) {

       sum += i;
```

```
        }
    }
    if (sum == num) {

        return 1;

    } else {
        return 0;
    }
}
int main() {
    int n;

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

    printf("Perfect numbers between 1 and %d are:\n", n);
    for (int i = 1; i <= n; i++) {

    if (isPerfect(i)) {

     printf("%d\n", i);

        }

    }

    return 0;

}
```

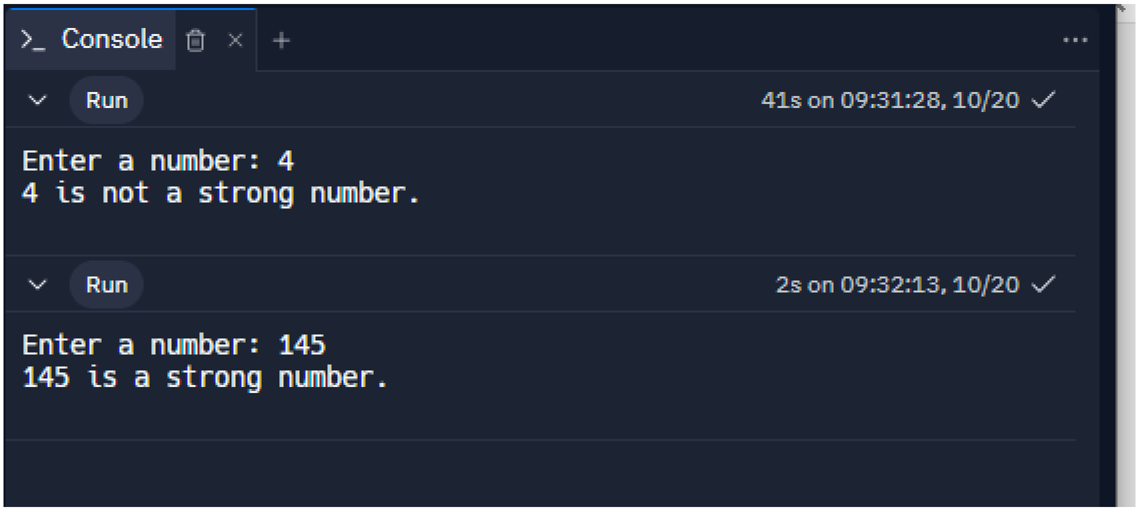


```
Enter the value of n: 62
Perfect numbers between 1 and 62 are:
6
28
```

```
Enter the value of n: 1000
Perfect numbers between 1 and 1000 are:
6
28
496
```

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

```
#include <stdio.h>
int factorial(int num) {
    if (num == 0 || num == 1) {
        return 1;
    }
    return num * factorial(num - 1);
}
int isStrong(int num) {
  int originalNum = num;

    int sum = 0;

        while (num > 0) {
        int digit = num % 10;
        sum += factorial(digit);
        num /= 10;

    }
    return sum == originalNum;
} int main() {
    int num;


    printf("Enter a number: ");
    scanf("%d", &num);


    if (isStrong(num)) {
        printf("%d is a strong number.\n", num);
    } else
 {
        printf("%d is not a strong number.\n", num);
    }
    return 0;
}
```



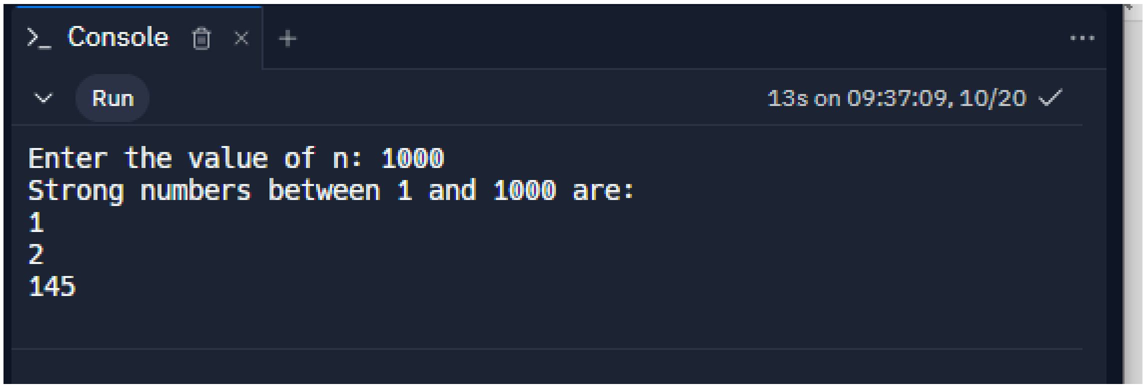
```
Enter a number: 4
4 is not a strong number.
```

```
Enter a number: 145
145 is a strong number.
```

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

```
#include <stdio.h>

int factorial(int num) {


  if (num == 0 || num == 1) {

    return 1;

  }

  return num * factorial(num - 1);

}

int isStrong(int num) {
  int originalNum = num;
  int sum = 0;

  while (num > 0) {
    int digit = num % 10;
    sum += factorial(digit);
    num /= 10;

  }
  return sum == originalNum;
}
int main() {
  int n;

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

  printf("Strong numbers between 1 and %d are:\n", n);
```

```
  for (int i = 1; i <= n; i++) {

    if (isStrong(i)) {
      printf("%d\n", i);
    }
  }
  return 0;
}
```

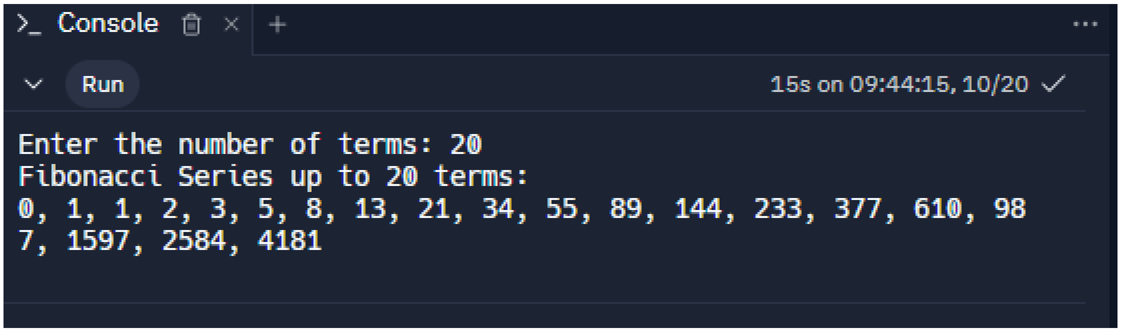


```
Enter the value of n: 1000
Strong numbers between 1 and 1000 are:
1
2
145
```

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

```
#include <stdio.h>
int main() {
```

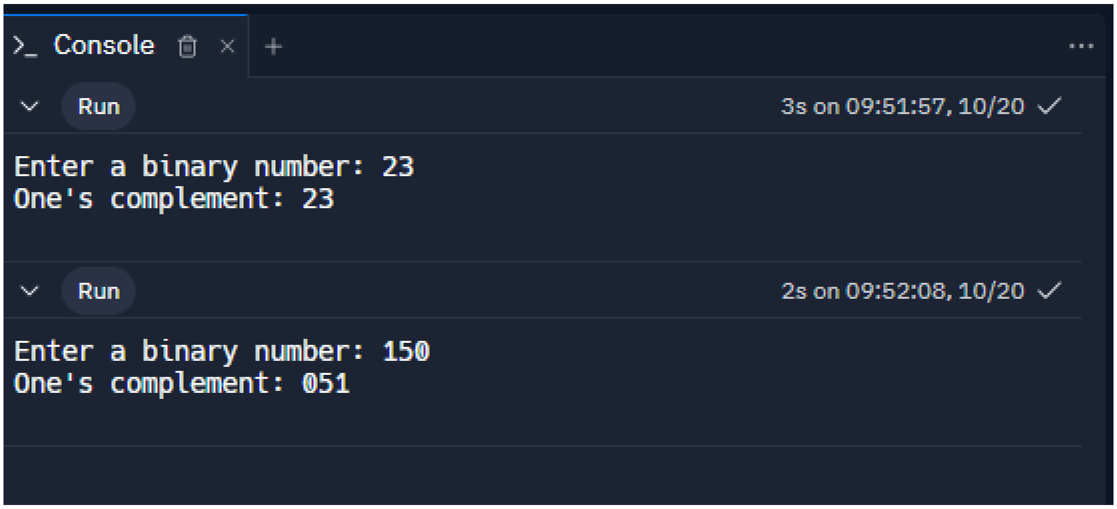
```
int n, first = 0, second = 1, next, i;


printf("Enter the number of terms: ");
scanf("%d", &n);


printf("Fibonacci Series up to %d terms:\n", n);


for (i = 0; i < n; i++) {
    if (i <= 1) {
        next = i;
   } else
  {
        next = first + second;
        first = second;
        second = next;
    }
    printf("%d", next);
    if (i < n - 1) {
            printf(", ");
        } else {
            printf("\n");
        }
    }
    return 0;
}
```



```
Enter the number of terms: 20
Fibonacci Series up to 20 terms:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 98
7, 1597, 2584, 4181
```

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

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


void onesComplement(char binary[]) {
    int length = strlen(binary);

 for (int i = 0; i < length; i++) {

        if (binary[i] == '0') {
            binary[i] = '1';
    }
     else if (binary[i] == '1') {
         binary[i] = '0';
        }
    }
}
int main() {

    char binary[100];

    printf("Enter a binary number: ");
    scanf("%s", binary);


    onesComplement(binary);


    printf("One's complement: %s\n", binary);

    return 0;

}
```



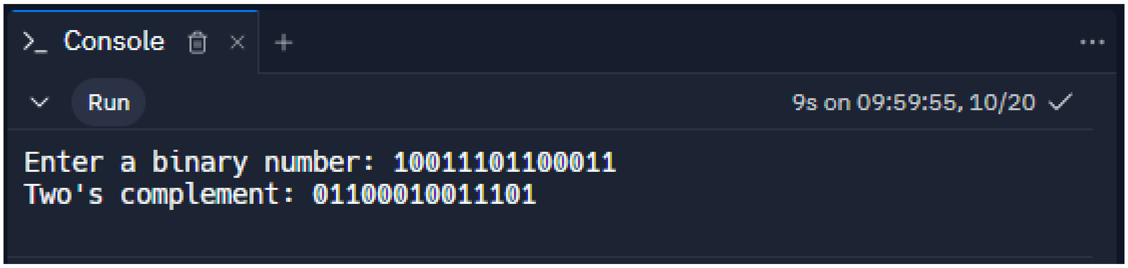
```
Enter a binary number: 23
One's complement: 23
```

```
Enter a binary number: 150
One's complement: 051
```

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

```
#include <stdio.h>

#include <string.h>
```

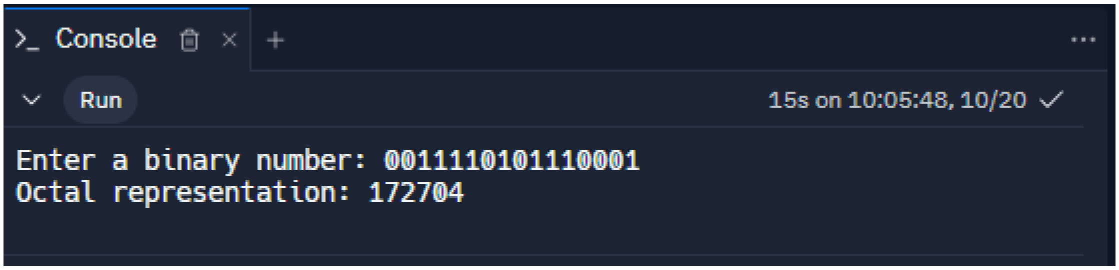
```
void onesComplement(char binary[]) {

    int length = strlen(binary);

    for (int i = 0; i < length; i++) {
        if (binary[i] == '0') {

            binary[i] = '1';

        } else if (binary[i] == '1') {
            binary[i] = '0';
        }
    }
}

void twosComplement(char binary[]) {

    onesComplement(binary);
    int length = strlen(binary);
    int carry = 1;


    for (int i = length - 1; i >= 0; i--) {
        if (binary[i] == '0' && carry == 1) {


            binary[i] = '1';
            carry = 0;
    }
       else if (binary[i] == '1' && carry == 1) {


            binary[i] = '0';
           carry = 1;
        }
 }
}
int main() {
    char binary[100];
    printf("Enter a binary number: ");
    scanf("%s", binary);

    twosComplement(binary);
    printf("Two's complement: %s\n", binary);

    return 0;
}
```



```
Enter a binary number: 10011101100011
Two's complement: 01100010011101
```

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

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

// Function to convert a binary digit to octal

int binaryToOctal(char binaryDigit[3]) {
    int octalDigit = 0;


    // Convert binaryDigit to decimal
    for (int i = 0; i < 3; i++)
 {


        octalDigit = octalDigit * 2 + (binaryDigit[i] - '0');
    }
    return octalDigit;
}
int main() {
    char binary[100];
    char binaryDigit[3];


    int length, i, j, k;


    printf("Enter a binary number: ");
    scanf("%s", binary);
    length = strlen(binary);

    if (length % 3 != 0) {

        int newLength = (length + 2) / 3 * 3; // Round up to the nearest multiple of 3
        for (i = length; i < newLength; i++) {

            binary[i] = '0';
        }
        binary[newLength] = '\0';
        length = newLength;
    }
    printf("Octal representation: ");

      for (i = 0; i < length; i += 3) {
        for (j = i, k = 0; j < i + 3; j++, k++) {

            binaryDigit[k] = binary[j];
        }
        binaryDigit[3] = '\0';
        int octalDigit = binaryToOctal(binaryDigit);
        printf("%d", octalDigit);
    }
    printf("\n");
    return 0;
}
```

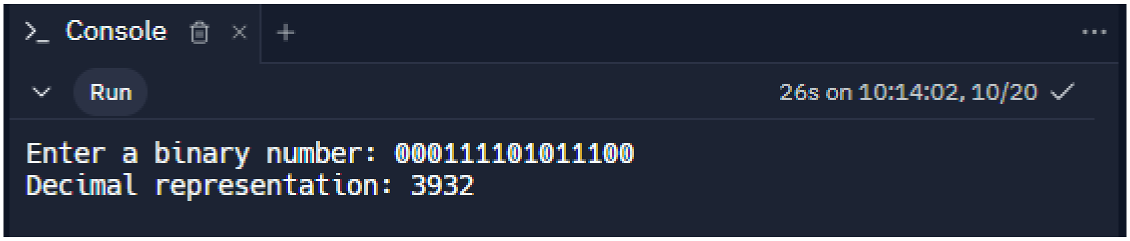


```
Enter a binary number: 0011110101110001
Octal representation: 172704
```

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

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

```
int binaryToDecimal(char binary[]) {

    int decimal = 0;
    int length = strlen(binary);

    for (int i = length - 1; i >= 0; i--) {
        if (binary[i] == '1') {

            decimal += 1 << (length - 1 - i);
        }
    }
    return decimal;
}
int main() {

    char binary[100];
    printf("Enter a binary number: ");
    scanf("%s", binary);

    int decimal = binaryToDecimal(binary);

    printf("Decimal representation: %d\n", decimal);
    return 0;
}
```

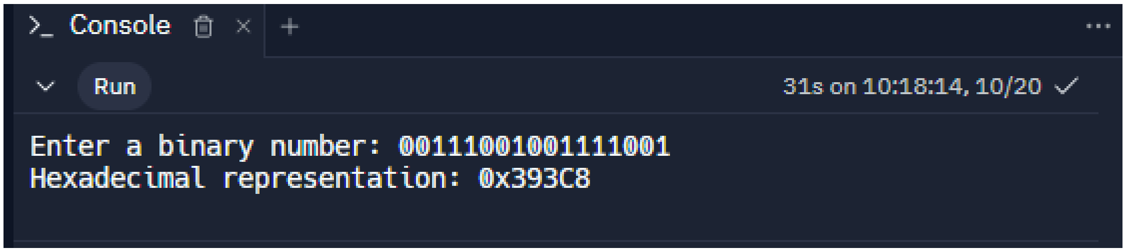


```
Enter a binary number: 000111101011100
Decimal representation: 3932
```

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

```
#include <stdio.h>
#include <string.h>
char binaryToHex(char binary[4]) {
    int decimal = 0;


    for (int i = 3; i >= 0; i--) {


        decimal += (binary[i] - '0') << (3 - i);
    }
    if (decimal >= 0 && decimal <= 9) {


        return (char)(decimal + '0');
    } else
 {
        return (char)(decimal - 10 + 'A');
    }
}
 int main() {
    char binary[100];
    char binaryDigit[4];
char hexadecimal[100];

int length, i, j, k;

printf("Enter a binary number: ");
scanf("%s", binary);

length = strlen(binary);
if (length % 4 != 0) {

    int newLength = (length + 3) / 4 * 4;

    for (i = length; i < newLength; i++) {

        binary[i] = '0';
    }
    binary[newLength] = '\0';
    length = newLength;
}
printf("Hexadecimal representation: 0x");


for (i = 0; i < length; i += 4) {
    for (j = i, k = 0; j < i + 4; j++, k++) {


        binaryDigit[k] = binary[j];
    }
    binaryDigit[4] = '\0';


    char hexDigit = binaryToHex(binaryDigit);


    printf("%c", hexDigit);


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

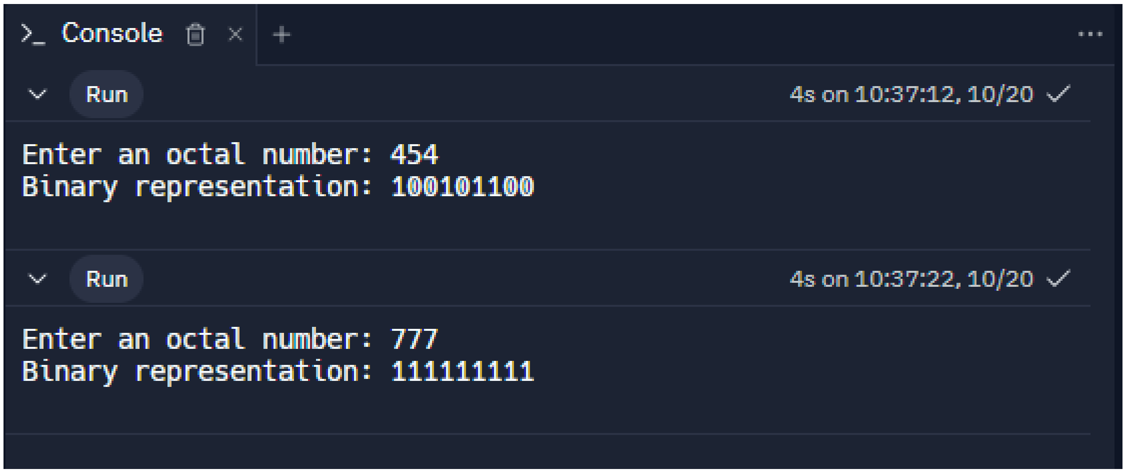
```
}
```



```
Enter a binary number: 00111001001111001
Hexadecimal representation: 0x393C8
```

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

```
#include <stdio.h>

#include <string.h>

char* octalToBinary(char octalDigit) {


    switch (octalDigit) {

        case '0': return "000";

        case '1': return "001";

        case '2': return "010";

        case '3': return "011";

        case '4': return "100";

        case '5': return "101";

        case '6': return "110";
        case '7': return "111";
        default: return NULL;      }
}
int main() {

    char octal[100];
    char binary[400];
    int length, i, j;

    printf("Enter an octal number: ");
    scanf("%s", octal);

    length = strlen(octal);

    for (i = 0, j = 0; i < length; i++) {

        char* binaryDigit = octalToBinary(octal[i]);
        if (binaryDigit == NULL) {

            printf("Invalid octal digit: %c\n", octal[i]);
            return 1;
        }
        strcpy(binary + j, binaryDigit);
        j += 3;
    }
    binary[j] = '\0';

    printf("Binary representation: %s\n", binary);

    return 0; }
```



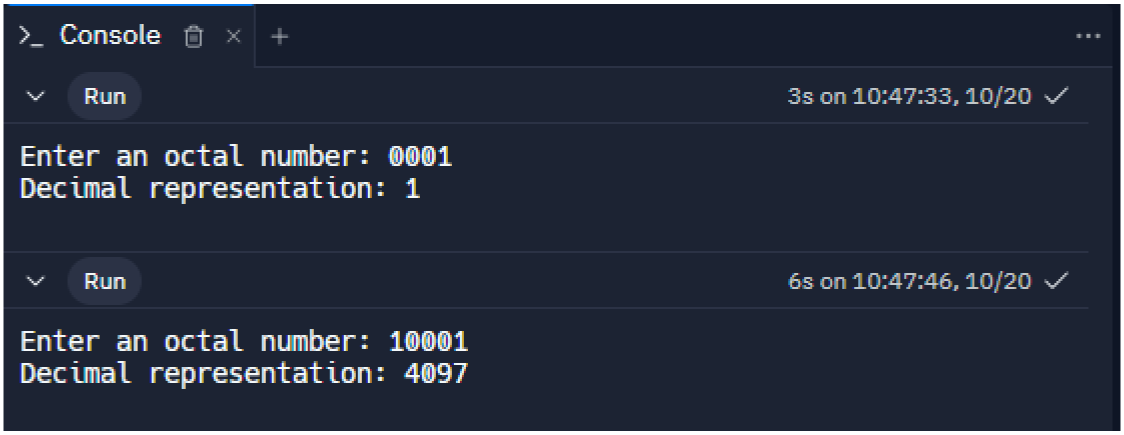
```
Enter an octal number: 454
Binary representation: 100101100
```

```
Enter an octal number: 777
Binary representation: 111111111
```

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

```
#include <stdio.h>

#include <string.h>


int octalToDecimal(char octal[]) {


    int decimal = 0;

    int length = strlen(octal);


    for (int i = 0; i < length; i++) {
        int octalDigit = octal[i] - '0';

        decimal = decimal * 8 + octalDigit;
    }
    return decimal;
}
int main()
 {
    char octal[100];

    printf("Enter an octal number: ");
    scanf("%s", octal);

    int decimal = octalToDecimal(octal);

    printf("Decimal representation: %d\n", decimal);
    return 0;
```

```
}
```



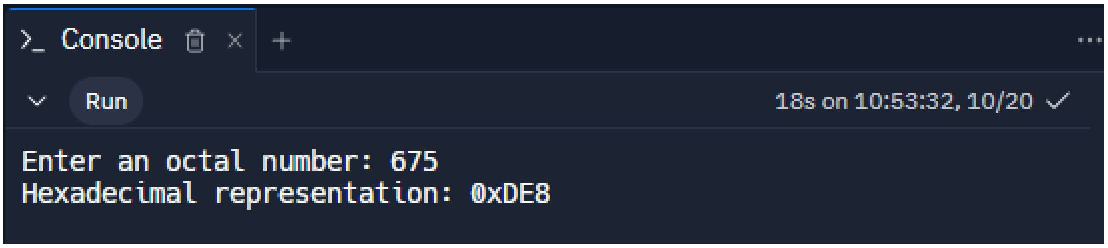
```
Enter an octal number: 0001
Decimal representation: 1
```

```
Enter an octal number: 10001
Decimal representation: 4097
```

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

```
#include <stdio.h>

#include <string.h>

char* octalToBinary(char octalDigit) {


    switch (octalDigit) {

        case '0': return "000";

        case '1': return "001";

        case '2': return "010";

        case '3': return "011";

        case '4': return "100";
        case '5': return "101";
        case '6': return "110";
        case '7': return "111";
        default: return NULL;
    }
}
char binaryToHex(char binary[4]) {

    int decimal = 0;

    for (int i = 3; i >= 0; i--)
    {
        decimal += (binary[i] - '0') << (3 - i);
    }
    if (decimal >= 0 && decimal <= 9) {
        return (char)(decimal + '0');
    } else
{
        return (char)(decimal - 10 + 'A');
    }
}
int main() {
    char octal[100];
    char binary[400];
    char hex[400];

    int length, i, j;
    printf("Enter an octal number: ");
    scanf("%s", octal);

    length = strlen(octal);
    for (i = 0, j = 0; i < length; i++) {
        char* binaryDigit = octalToBinary(octal[i]);
        if (binaryDigit == NULL) {
            printf("Invalid octal digit: %c\n", octal[i]);
            return 1;
        }
        strcpy(binary + j, binaryDigit);
        j += 3;
    }
    binary[j] = '\0';
    if (j % 4 != 0) {
        int newLength = (j + 3) / 4 * 4;
        for (i = j; i < newLength; i++) {
            binary[i] = '0';
        }
        binary[newLength] = '\0';
        j = newLength;
    }
    printf("Hexadecimal representation: 0x");
    for (i = 0; i < j; i += 4) {
        char binaryDigit[5];
        for (int k = 0; k < 4; k++) {
            binaryDigit[k] = binary[i + k];
        }
        binaryDigit[4] = '\0';
        char hexDigit = binaryToHex(binaryDigit);
```

```
        printf("%c", hexDigit);
    }
    printf("\n");

    return 0;
}
```

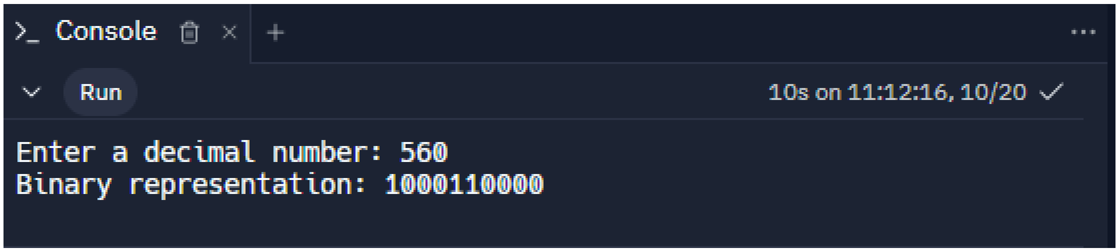


```
Enter an octal number: 675
Hexadecimal representation: 0xDE8
```

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

```
#include <stdio.h>


void decimalToBinary(int decimal)
 {
    int binary[32];
    int index = 0;

    if (decimal == 0)
```

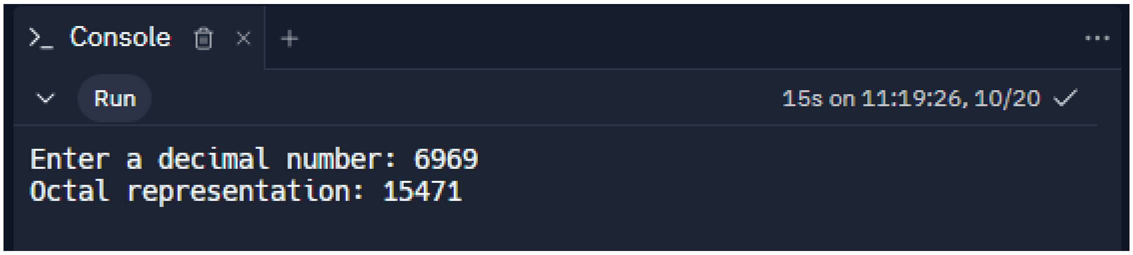
```
{
        printf("Binary representation: 0\n");
        return;
    }
    while (decimal > 0) {
        binary[index] = decimal % 2;
        decimal = decimal / 2;
        index++;

    }
    printf("Binary representation: ");

    for (int i = index - 1; i >= 0; i--) {

        printf("%d", binary[i]);
    }
    printf("\n");
}
int main() {
    int decimal;

    printf("Enter a decimal number: ");

    scanf("%d", &decimal);



    decimalToBinary(decimal);

    return 0;

}
```



```
Enter a decimal number: 560
Binary representation: 1000110000
```

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

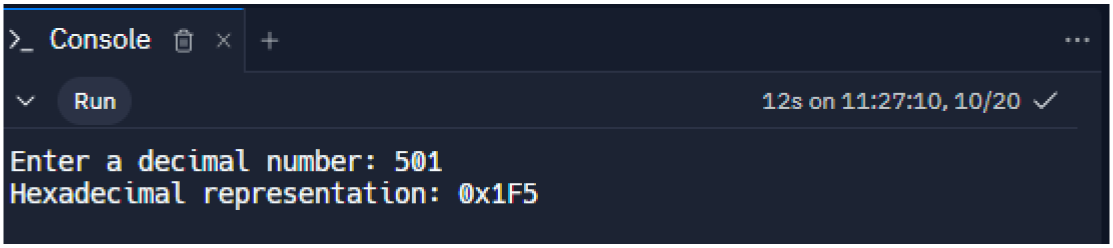
```
#include <stdio.h>

void decimalToOctal(int decimal) {



    int octal[32];

    int index = 0;

    if (decimal == 0) {
        printf("Octal representation: 0\n");

        return;
    }
    while (decimal > 0) {
        octal[index] = decimal % 8;
        decimal = decimal / 8;
        index++;
    }
    printf("Octal representation: ");

    for (int i = index - 1; i >= 0; i--) {

        printf("%d", octal[i]);
    }
    printf("\n");
}
int main()
 {
    int decimal;


    printf("Enter a decimal number: ");
    scanf("%d", &decimal);


    decimalToOctal(decimal);
    return 0;
}
```



```
Enter a decimal number: 6969
Octal representation: 15471
```

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

```
#include <stdio.h>
char decimalToHexDigit(int decimalDigit) {
    if (decimalDigit >= 0 && decimalDigit <= 9)
{
        return (char)('0' + decimalDigit);
    }
 else
{
        return (char)('A' + (decimalDigit - 10));
    }
}
void decimalToHexadecimal(int decimal) {

    char hexadecimal[32];

    int index = 0;
    if (decimal == 0)
{
        printf("Hexadecimal representation: 0\n");
        return;
    }
    while (decimal > 0) {
        int remainder = decimal % 16;
        hexadecimal[index] = decimalToHexDigit(remainder);
        decimal = decimal / 16;
        index++;
    }
    printf("Hexadecimal representation: 0x");

    for (int i = index - 1; i >= 0; i--) {

        printf("%c", hexadecimal[i]);
    }
    printf("\n");
}
int main() {
    int decimal;

    printf("Enter a decimal number: ");
    scanf("%d", &decimal);

    decimalToHexadecimal(decimal);
    return 0;

}
```



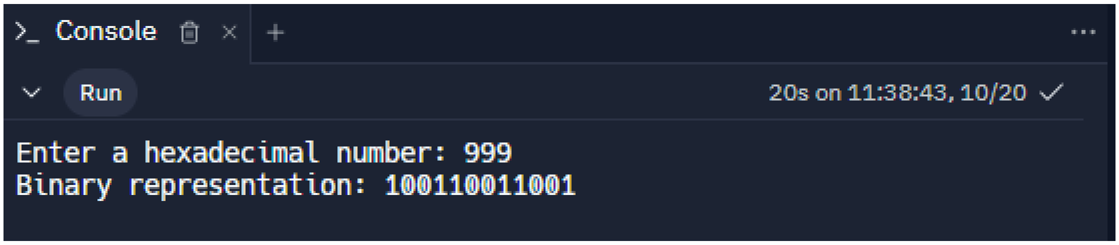
```
Enter a decimal number: 501
Hexadecimal representation: 0x1F5
```

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

```
#include <stdio.h>

#include <string.h>

char* hexToBinary(char hexDigit) {

    switch (hexDigit) {
        case '0': return "0000";
        case '1': return "0001";
        case '2': return "0010";
        case '3': return "0011";
        case '4': return "0100";
        case '5': return "0101";
        case '6': return "0110";
        case '7': return "0111";
        case '8': return "1000";
        case '9': return "1001";
        case 'A': case 'a': return "1010";
        case 'B': case 'b': return "1011";
        case 'C': case 'c': return "1100";
        case 'D': case 'd': return "1101";
        case 'E': case 'e': return "1110";
        case 'F': case 'f': return "1111";
        default: return NULL;

    }
}
int main()
{
    char hexadecimal[100];
    char binary[400];
```

```
}

 int length, i, j;

printf("Enter a hexadecimal number: ");
 scanf("%s", hexadecimal);

length = strlen(hexadecimal);

 printf("Binary representation: ");

 for (i = 0, j = 0; i < length; i++) {

     char* binaryDigit = hexToBinary(hexadecimal[i]);

     if (binaryDigit == NULL) {

     printf("Invalid hexadecimal digit: %c\n", hexadecimal[i]);

 return 1;
 }
```

```
    printf("%s", binaryDigit);
   }
  printf("\n");

  return 0;
}
```



```
Enter a hexadecimal number: 999
Binary representation: 100110011001
```

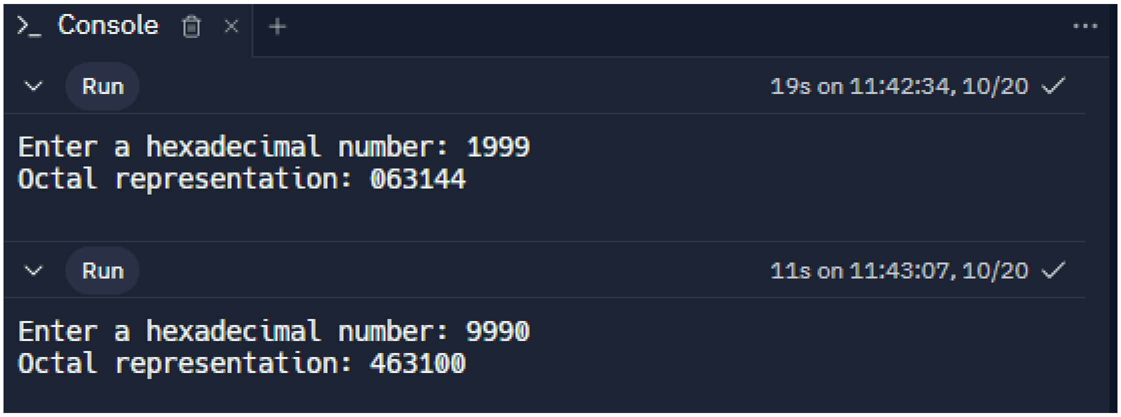
70. Write a C program to convert Hexadecimal to Octal number system.

```
#include <stdio.h>

#include <string.h>

char* hexToBinary(char hexDigit) {

    switch (hexDigit) {
```

```
    case '0': return "0000";

    case '1': return "0001";

    case '2': return "0010";

    case '3': return "0011";

    case '4': return "0100";

    case '5': return "0101";

    case '6': return "0110";

    case '7': return "0111";

    case '8': return "1000";

    case '9': return "1001";

    case 'A': case 'a': return "1010";

    case 'B': case 'b': return "1011";

    case 'C': case 'c': return "1100";

    case 'D': case 'd': return "1101";

    case 'E': case 'e': return "1110";

    case 'F': case 'f': return "1111";


    default: return NULL;

}
```

```
}
char binaryToOctal(char binary[3]) {

    int decimal = 0;

    for (int i = 2; i >= 0; i--) {

        decimal += (binary[i] - '0') << (2 - i);
    }
    return (char)('0' + decimal);
}
int main() {
    char hexadecimal[100];
    char binary[400];
    char octal[400];

    int length, i, j, k;

    printf("Enter a hexadecimal number: ");
```

```
scanf("%s", hexadecimal);
length = strlen(hexadecimal);
printf("Octal representation: ");
for (i = 0, j = 0; i < length; i++) {
    char* binaryDigit = hexToBinary(hexadecimal[i]);
    if (binaryDigit == NULL) {
        printf("Invalid hexadecimal digit: %c\n", hexadecimal[i]);
        return 1;
    }
    strcpy(binary + j, binaryDigit);
    j += 4;
}
if (j % 3 != 0) {

    int newLength = (j + 2) / 3 * 3;

    for (k = j; k < newLength; k++) {
        binary[k] = '0';
    }

    binary[newLength] = '\0';

    j = newLength;
}
for (i = 0; i < j; i += 3) {

    char binaryDigit[4];
    for (k = 0; k < 3; k++) {
            binaryDigit[k] = binary[i + k];
        }
        binaryDigit[3] = '\0';

        char octalDigit = binaryToOctal(binaryDigit);
        printf("%c", octalDigit);
    }
  printf("\n");
   return 0;
}
```



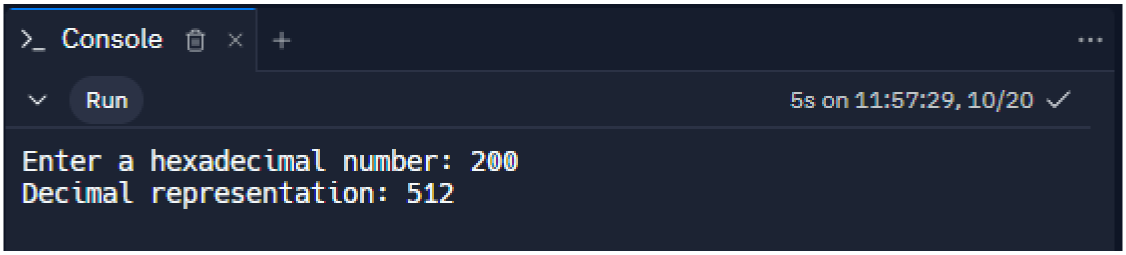
```
Enter a hexadecimal number: 1999
Octal representation: 063144
```

```
Enter a hexadecimal number: 9990
Octal representation: 463100
```

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

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

int hexToDecimalDigit(char hexDigit) {
    if (hexDigit >= '0' && hexDigit <= '9') {
        return hexDigit - '0';

    } else if (hexDigit >= 'A' && hexDigit <= 'F') {
        return hexDigit - 'A' + 10;

    } else if (hexDigit >= 'a' && hexDigit <= 'f') {
        return hexDigit - 'a' + 10;

    } else {

        return -1;
    }
}
```

```
int hexToDecimal(char hexadecimal[]) {

    int decimal = 0;

    int length = strlen(hexadecimal);

    for (int i = 0; i < length; i++) {

        int decimalDigit = hexToDecimalDigit(hexadecimal[i]);

        if (decimalDigit == -1) {

            printf("Invalid hexadecimal digit: %c\n", hexadecimal[i]);

            return -1;

        }

        decimal = decimal * 16 + decimalDigit;
    }

    return decimal;
}
int main() {
    char hexadecimal[100];
    printf("Enter a hexadecimal number: ");
    scanf("%s", hexadecimal);

    int decimal = hexToDecimal(hexadecimal);
    if (decimal != -1) {
        printf("Decimal representation: %d\n", decimal);
    }
 return 0;
}
```

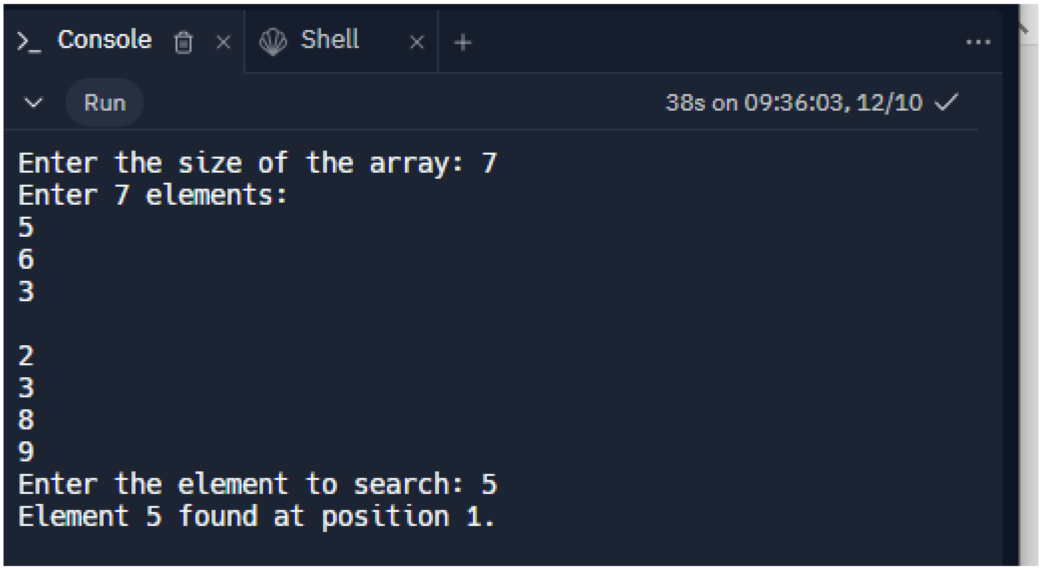


```
Enter a hexadecimal number: 200
Decimal representation: 512
```

72. **Write a C program to search for an element in an array. Display the position of the element.**

```
#include <stdio.h>

// Function to search for an element in the array
int searchElement(int arr[], int n, int key) {
 for (int i = 0; i < n; i++) {
  if (arr[i] == key) {
   return i; // Return the index if element is found
      }
    }
    return -1; // Return -1 if element is not found
}

int main() {
    int n, key;

    // Input the size of the array
    printf("Enter the size of the array: ");
    scanf("%d", &n);
```

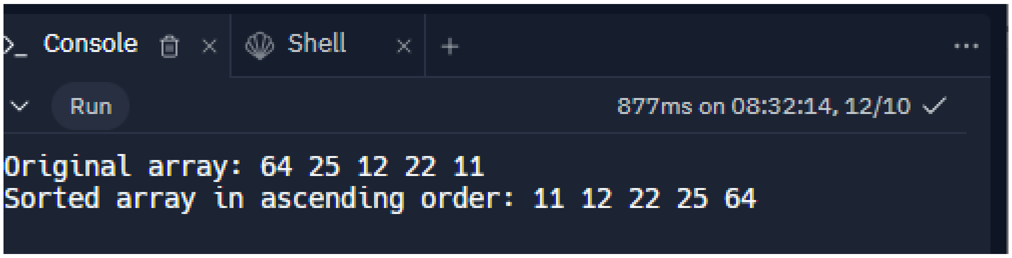
```c
    int arr[n];

    // Input array elements
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Input the element to search
    printf("Enter the element to search: ");
    scanf("%d", &key);

    // Search for the element in the array
    int position = searchElement(arr, n, key);

    // Display the result
    if (position != -1) {
        printf("Element %d found at position %d.\n", key, position
+ 1); // Adding 1 to convert index to position
    } else {
        printf("Element %d not found in the array.\n", key);
    }

    return 0;
}
```



```
Enter the size of the array: 7
Enter 7 elements:
5
6
3

2
3
8
9
Enter the element to search: 5
Element 5 found at position 1.
```

73. . Write a C program to sort the elements of an array in ascending order.

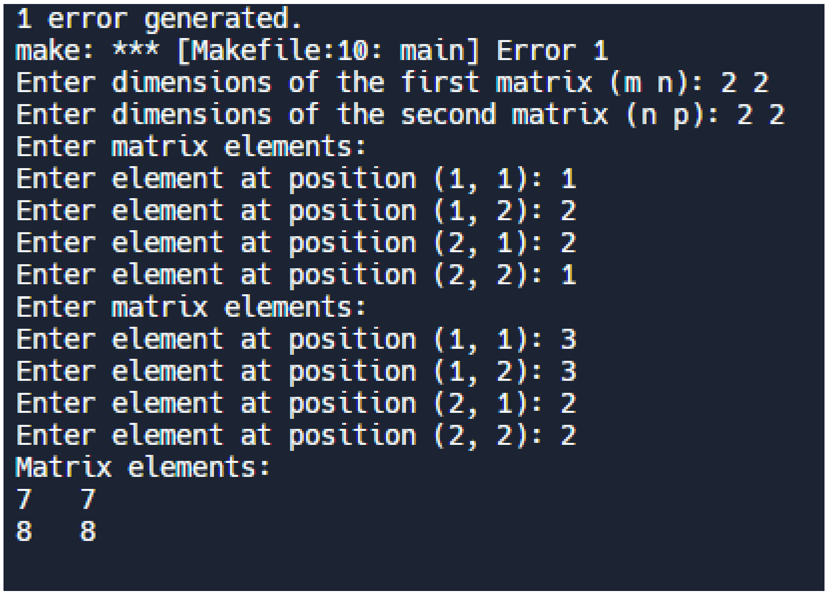
```
#include <stdio.h>

void swap(int a,b,c) {
    c = a;
     a = b;
     b = c;
}

void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                swap(&arr[j], &arr[j + 1]);
    }
    }
   }
  }

void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr) / sizeof(arr[0]);

    printf("Original array: ");
    printArray(arr, n);

    bubbleSort(arr, n);

    printf("Sorted array in ascending order: ");
    printArray(arr, n);

    return 0;
}
```



```
Original array: 64 25 12 22 11
Sorted array in ascending order: 11 12 22 25 64
```

74. Consider two matrices of the size m and n. Implement matrix multiplication operation and display results using functions. Write three functions 1) Read matrix elements 2) Matrix Multiplication 3) Print matrix elements

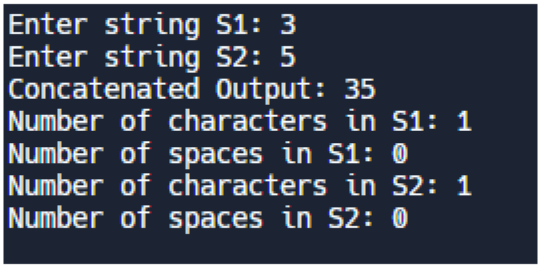
```
#include <stdio.h>

// Function to read matrix elements
void readMatrix(int rows, int cols, int matrix[][cols]) {
  printf("Enter matrix elements:\n");
  for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
      printf("Enter element at position (%d, %d): ", i + 1, j + 1);
      scanf("%d", &matrix[i][j]);
  }
  }
  }

  // Function to perform matrix multiplication

void multiplyMatrices(int m, int n, int p, int matrix1[][n], int
matrix2[][p], int result[][p]) {
  for (int i = 0; i < m; ++i) {
  for (int j = 0; j < p; ++j) {
    result[i][j] = 0;
    for (int k = 0; k < n; ++k) {
    result[i][j] += matrix1[i][k] * matrix2[k][j];
}
  }
  }
 }
// Function to print matrix elements
void printMatrix(int rows, int cols, int matrix[][cols]) {
    printf("Matrix elements:\n");
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
       }
       }
     int main() {
    int m, n, p;
  // Read dimensions of matrices
    printf("Enter dimensions of the first matrix (m n): ");
    scanf("%d %d", &m, &n);

    printf("Enter dimensions of the second matrix (n p): ");
    scanf("%d %d", &n, &p);

    // Check if matrices can be multiplied
    if (n <= 0 || m <= 0 || p <= 0) {
        printf("Invalid dimensions for matrix multiplication.\n");
        return 1;
    }
    int matrix1[m][n], matrix2[n][p], result[m][p];

    // Read matrix elements
    readMatrix(m, n, matrix1);
    readMatrix(n, p, matrix2);

    // Perform matrix multiplication
    multiplyMatrices(m, n, p, matrix1, matrix2, result);

    // Print result matrix
    printMatrix(m, p, result);
  return 0;
}
```



```
1 error generated.
make: *** [Makefile:10: main] Error 1
Enter dimensions of the first matrix (m n): 2 2
Enter dimensions of the second matrix (n p): 2 2
Enter matrix elements:
Enter element at position (1, 1): 1
Enter element at position (1, 2): 2
Enter element at position (2, 1): 2
Enter element at position (2, 2): 1
Enter matrix elements:
Enter element at position (1, 1): 3
Enter element at position (1, 2): 3
Enter element at position (2, 1): 2
Enter element at position (2, 2): 2
Matrix elements:
7   7
8   8
```

75. Consider two strings S1 and S2. Develop a C Program for the following operations.a) Display a concatenated output of S1 and S2b) Count the number of characters and empty spaces in S1and S2.

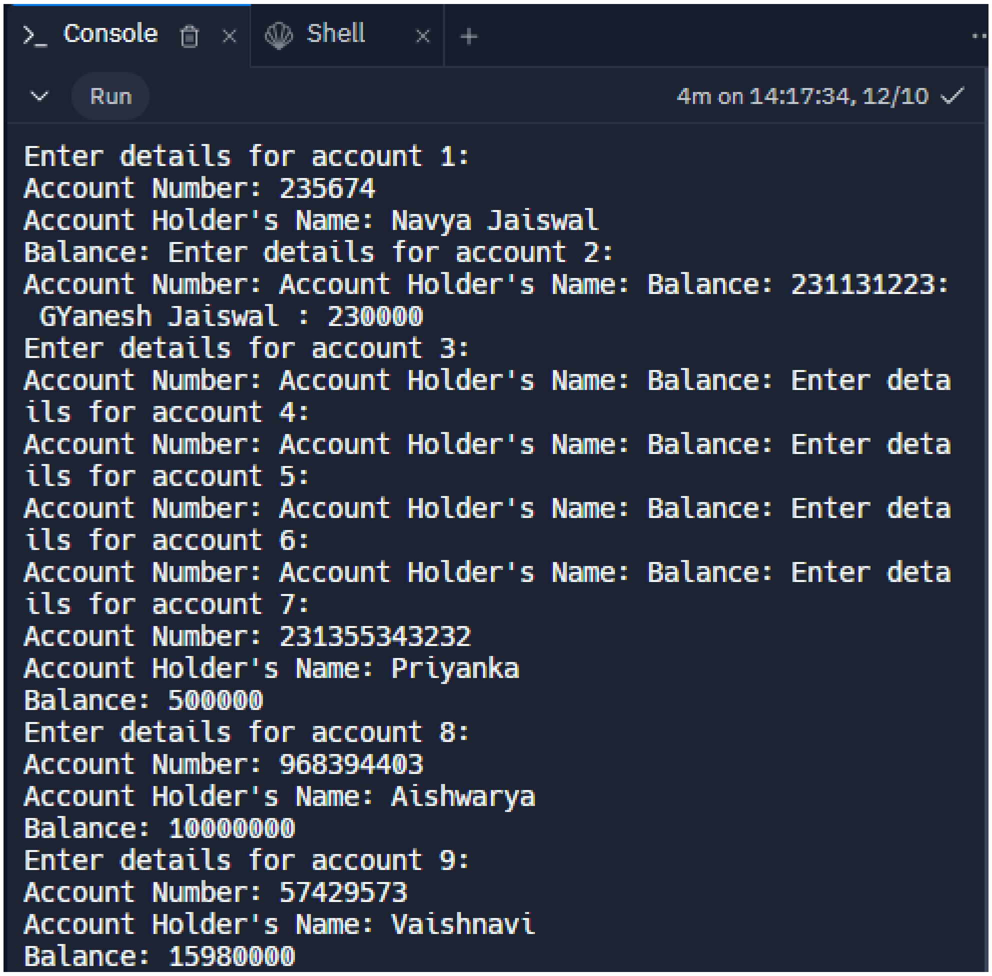
```
#include <stdio.h>

#include <string.h>

int main() {

    char S1[100], S2[100];

// Input the strings

    printf("Enter string S1: ");

    gets(S1); // Note: gets() is used for simplicity; it's not recommended for real-world use.

    printf("Enter string S2: ");

    gets(S2);

    // Concatenate and display the output of S1 and S2

    printf("Concatenated Output: %s%s\n", S1, S2);


    // Count the number of characters and empty spaces in S1 and S2

    int charCountS1 = 0, spaceCountS1 = 0;

    int charCountS2 = 0, spaceCountS2 = 0;

    for (int i = 0; i < strlen(S1); ++i) {

        if (S1[i] == ' ') {

            spaceCountS1++;

        } else {
            charCountS1++;
        }
    }
    for (int i = 0; i < strlen(S2); ++i) {
        if (S2[i] == ' ') {
            spaceCountS2++;
        } else {
            charCountS2++;
        }
    }
    // Display the counts
    printf("Number of characters in S1: %d\n", charCountS1);
    printf("Number of spaces in S1: %d\n", spaceCountS1);
    printf("Number of characters in S2: %d\n", charCountS2);
    printf("Number of spaces in S2: %d\n", spaceCountS2);
    return 0;
}
```



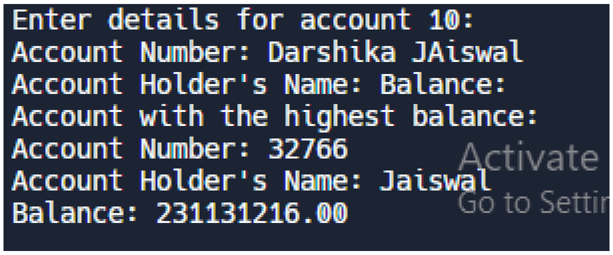
```
Enter string S1: 3
Enter string S2: 5
Concatenated Output: 35
Number of characters in S1: 1
Number of spaces in S1: 0
Number of characters in S2: 1
Number of spaces in S2: 0
```

Q76. Consider details of a bank account with the fields account number, account holder’s name, and balance. Write a program to read 10 people’s details and display the record with the highest bank balance.

```
#include <stdio.h>
// Structure to represent a bank account
struct BankAccount {
    int accountNumber;
    char accountHolderName[50];
    float balance;
};
int main() {
    // Declare an array of BankAccount structures to store details of 10 accounts
    struct BankAccount accounts[10];
    // Read details of 10 accounts
    for (int i = 0; i < 10; i++) {
        printf("Enter details for account %d:\n", i + 1);
        printf("Account Number: ");
        scanf("%d", &accounts[i].accountNumber);

        printf("Account Holder's Name: ");
        scanf("%s", accounts[i].accountHolderName);

        printf("Balance: ");
        scanf("%f", &accounts[i].balance);

    }

    // Find the record with the highest bank balance

    float maxBalance = accounts[0].balance;

    int maxIndex = 0;

    for (int i = 1; i < 10; i++) {

        if (accounts[i].balance > maxBalance) {

            maxBalance = accounts[i].balance;

            maxIndex = i;

        }

    }

    // Display the record with the highest bank balance

    printf("\nAccount with the highest balance:\n");

    printf("Account Number: %d\n", accounts[maxIndex].accountNumber);

    printf("Account Holder's Name: %s\n",
accounts[maxIndex].accountHolderName);

    printf("Balance: %.2f\n", accounts[maxIndex].balance);

  return 0;

}
```



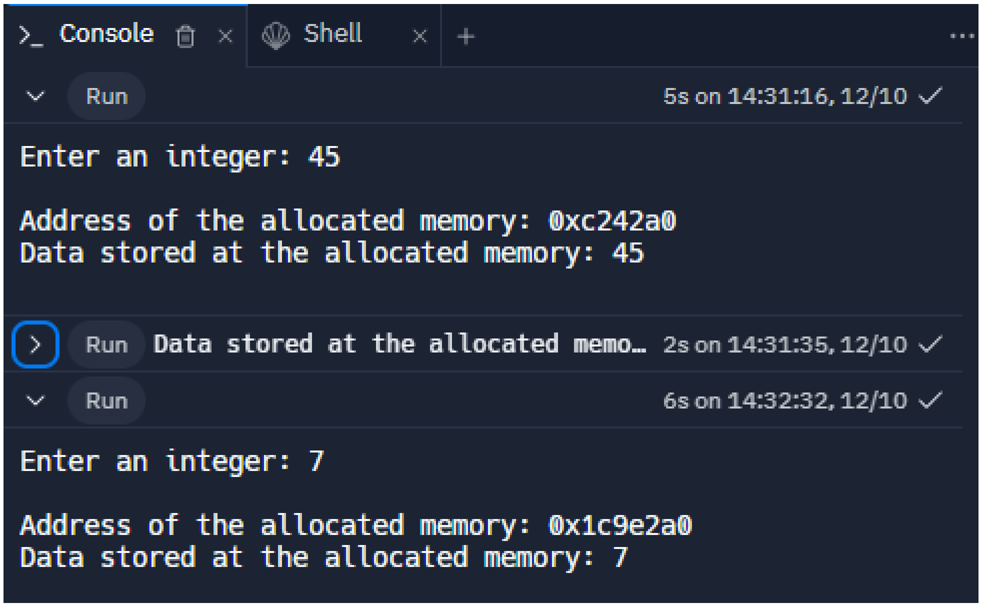
```
Enter details for account 1:
Account Number: 235674
Account Holder's Name: Navya Jaiswal
Balance: Enter details for account 2:
Account Number: Account Holder's Name: Balance: 231131223:
 GYanesh Jaiswal : 230000
Enter details for account 3:
Account Number: Account Holder's Name: Balance: Enter deta
ils for account 4:
Account Number: Account Holder's Name: Balance: Enter deta
ils for account 5:
Account Number: Account Holder's Name: Balance: Enter deta
ils for account 6:
Account Number: Account Holder's Name: Balance: Enter deta
ils for account 7:
Account Number: 231355343232
Account Holder's Name: Priyanka
Balance: 500000
Enter details for account 8:
Account Number: 968394403
Account Holder's Name: Aishwarya
Balance: 10000000
Enter details for account 9:
Account Number: 57429573
Account Holder's Name: Vaishnavi
Balance: 15980000
```



```
Enter details for account 10:
Account Number: Darshika JAiswal
Account Holder's Name: Balance:
Account with the highest balance:
Account Number: 32766
Account Holder's Name: Jaiswal
Balance: 231131216.00
```

Q77. Q9. Write a C program to demonstrate the use of & and * operators using pointers. Create and free a memory location for an integer. Display the address and data stored at the location.

```
#include <stdio.h>

int main() {

    // Declare a pointer to an integer

    int *ptr;
    // Dynamically allocate memory for an integer
    ptr = (int *)malloc(sizeof(int));
    // Check if memory allocation was successful
    if (ptr == NULL) {
        printf("Memory allocation failed.\n");
        return 1; // Exit with an error code
    }
    // Input data at the allocated memory location
    printf("Enter an integer: ");
    scanf("%d", ptr);
    // Display the address and data stored at the allocated memory location
    printf("\nAddress of the allocated memory: %p\n", (void *)ptr);
    printf("Data stored at the allocated memory: %d\n", *ptr);
    // Free the allocated memory
    free(ptr);
    return 0;
}
```



```
Enter an integer: 45

Address of the allocated memory: 0xc242a0
Data stored at the allocated memory: 45
```

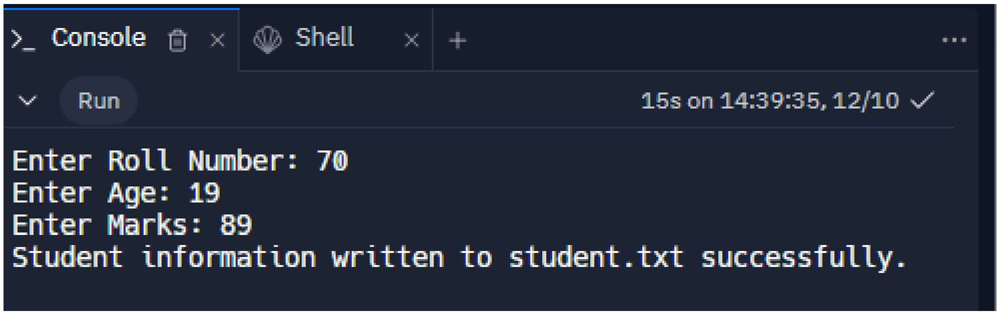
```
Enter an integer: 7

Address of the allocated memory: 0x1c9e2a0
Data stored at the allocated memory: 7
```

Q78. Q10. Write a program to create a file called student.txt and store information about a student in terms of roll no, age, and marks.

```
#include <stdio.h>

// Structure to represent a student

struct Student {

    int rollNumber;

    int age;

    float marks;

};

int main() {

    // Declare a file pointer

    FILE *filePointer;

    // Declare a variable of type struct Student

    struct Student student;
```

```
// Open the file for writing

filePointer = fopen("student.txt", "w");

// Check if the file is opened successfully

if (filePointer == NULL) {

    printf("Error opening the file.\n");

    return 1; // Exit with an error code

}

// Input student information

printf("Enter Roll Number: ");

scanf("%d", &student.rollNumber);

printf("Enter Age: ");

scanf("%d", &student.age);

printf("Enter Marks: ");

scanf("%f", &student.marks);

// Write student information to the file

fprintf(filePointer, "Roll Number: %d\n", student.rollNumber);

fprintf(filePointer, "Age: %d\n", student.age);

fprintf(filePointer, "Marks: %.2f\n", student.marks);

// Close the file

fclose(filePointer);

printf("Student information written to student.txt successfully.\n");

return 0;
```

}



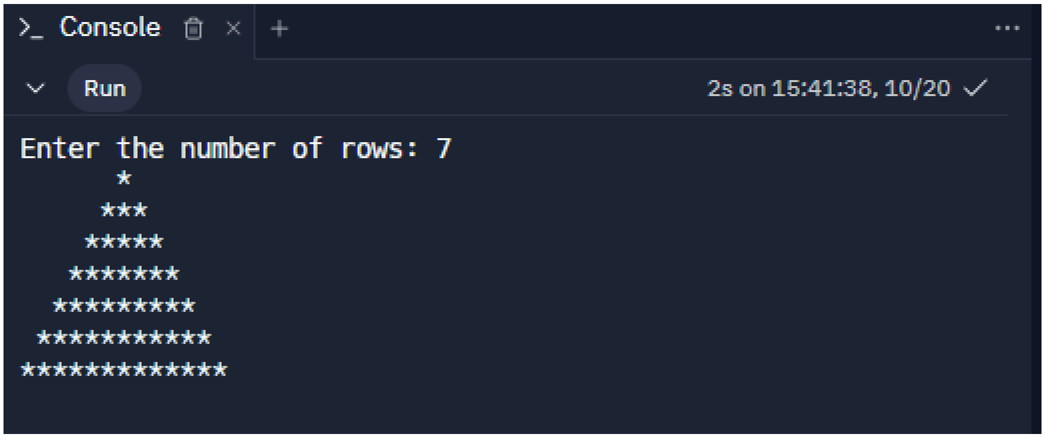
```
Enter Roll Number: 70
Enter Age: 19
Enter Marks: 89
Student information written to student.txt successfully.
```

Pattern Exercises

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

Pyramid star pattern.

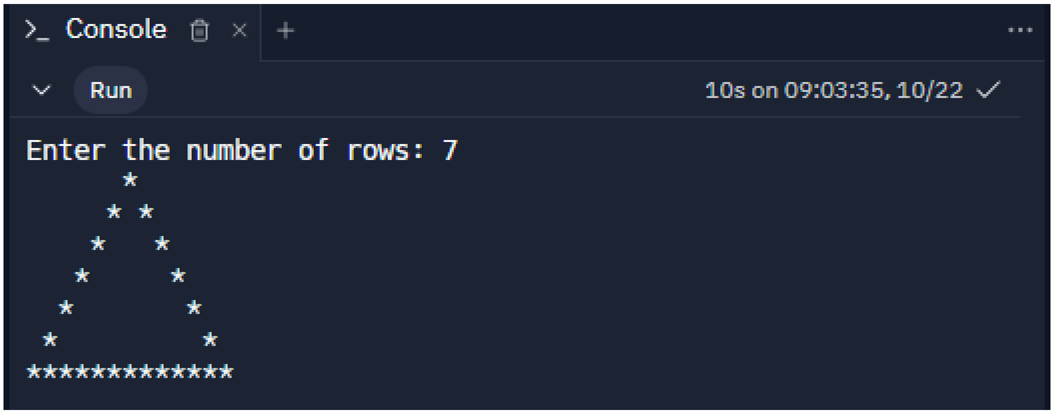
```
#include <stdio.h>


int main() {


    int rows, i, j, space;


  printf("Enter the number of rows: ");
  scanf("%d", &rows);


  for (i = 1; i <= rows; i++) {
      for (space = 1; space <= rows - i; space++) {
       printf(" ");
     }
     for (j = 1; j <= 2 * i - 1; j++) {
       printf("*");
       }
    printf("\n");
    }
   return 0;
}
```



```
Enter the number of rows: 7
      *
     ***
    *****
   *******
  *********
 ***********
*************
```

Hollow Pyramid Star Print.

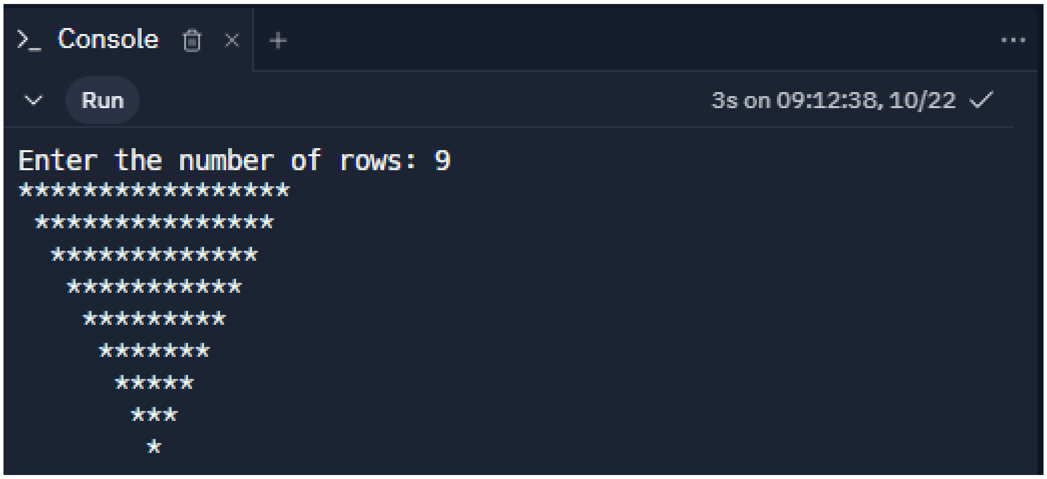
```
#include <stdio.h>


int main() {
 int rows, i, j, space;


 printf("Enter the number of rows: ");
  scanf("%d", &rows);


    for (i = 1; i <= rows; i++) {
     for (space = 1; space <= rows - i; space++) {
        printf(" ");
       }
      if (i == 1 || i == rows) {


        for (j = 1; j <= 2 * i - 1; j++) {
      printf("*");
    }
   }
        else
   {
     printf("*");

     for (j = 1; j <= 2 * i - 3; j++) {

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



```
Enter the number of rows: 7
      *
     * *
    *   *
   *     *
  *       *
 *         *
*************
```

Reverse pyramid star pattern .

```
#include <stdio.h>
int main() {


    int rows, i, j, space;
    printf("Enter the number of rows: ");
    scanf("%d", &rows);


    for (i = rows; i >= 1; i--) {
    for (space = 1; space <= rows - i; space++) {
     printf(" ");
  }
    for (j = 1; j <= 2 * i - 1; j++) {
     printf("*");
   }
    printf("\n");
 }
  return 0;
}
```



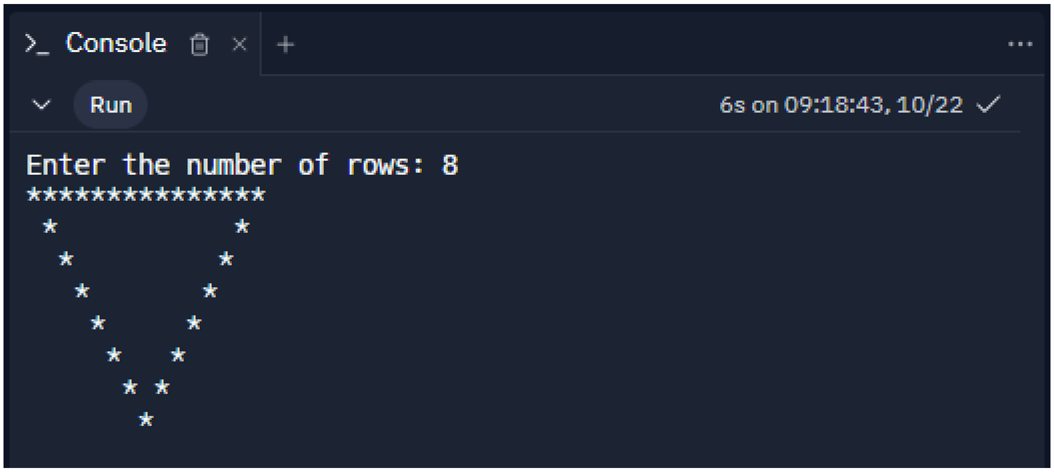
```
Enter the number of rows: 9
*****************
 ***************
  *************
   ***********
    *********
     *******
      *****
       ***
        *
```

Inverted hollow pyramid star pattern .

```
#include <stdio.h>


int main() {

int rows, i, j, space;


  printf("Enter the number of rows: ");

  scanf("%d", &rows);


    for (i = rows; i >= 1; i--) {

      for (space = 1; space <= rows - i; space++) {
```

```
     printf(" ");
  }
    if (i == 1 || i == rows) {
    for (j = 1; j <= 2 * i - 1; j++) {

  printf("*");
 }
 }
else
{
      printf("*");

    for (j = 1; j <= 2 * i - 3; j++) {

      printf(" ");
 }
```

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



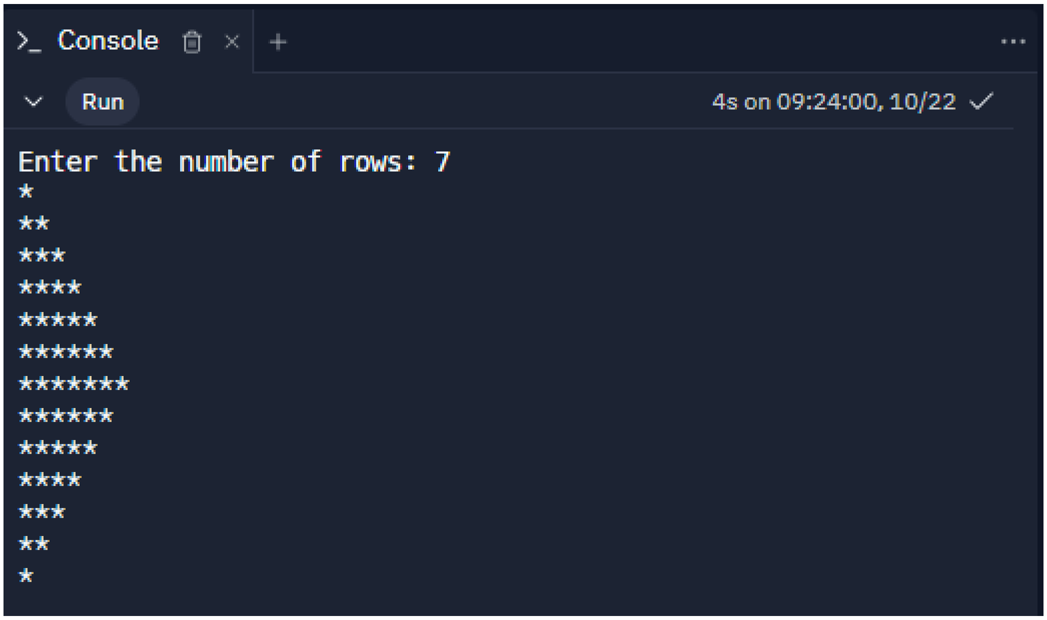
```
Enter the number of rows: 8
***************
 *           *
  *         *
   *       *
    *     *
     *   *
      * *
       *
```

Half diamond star pattern.

```
#include <stdio.h>




int main() {
```

```
int n, i, j;


printf("Enter the number of rows: ");
 scanf("%d", &n);


for (i = 1; i <= n; i++)
  {
 for (j = 1; j <= i; j++)
{
 printf("*");
 }
 printf("\n");
}
for (i = n - 1; i >= 1; i--) {
 for (j = 1; j <= i; j++) {
   printf("*");
 }
```

```
    printf("\n");

  }

  return 0;

}
```



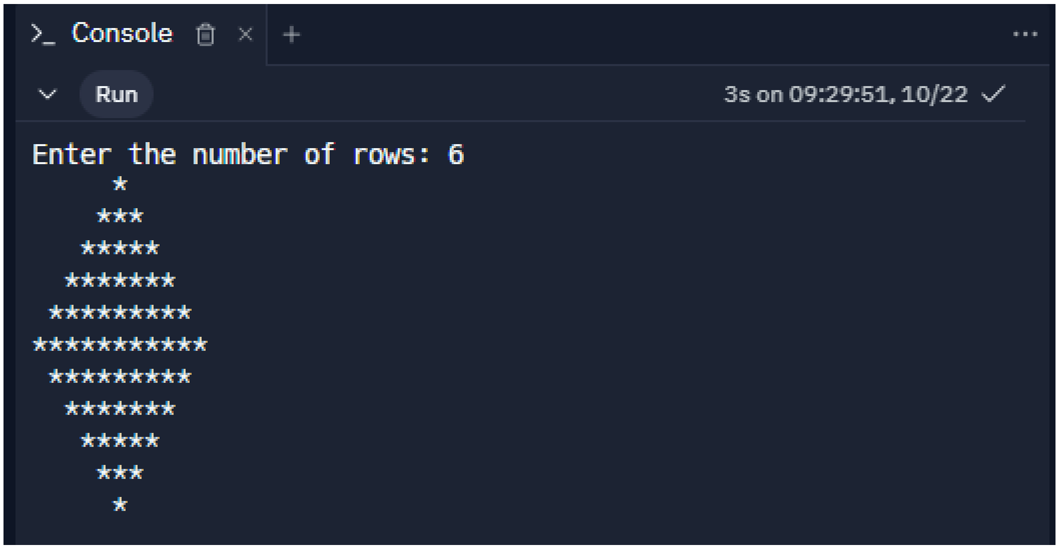
```
Enter the number of rows: 7
*
**
***
****
*****
******
*******
******
*****
****
***
**
*
```

Mirrored diamond star pattern.

```
#include <stdio.h>


int main() {

  int n, i, j, k;


  printf("Enter the number of rows: ");
```

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


  for (i = 1; i <= n; i++) {


      for (j = 1; j <= n - i; j++) {

    printf(" ");

  }

   for (k = 1; k <= 2 * i - 1; k++) {

     printf("*");

}

printf("\n");

  }

  for (i = n - 1; i >= 1; i--) {

  for (j = 1; j <= n - i; j++)

   {

    printf(" ");

}
```

```
    for (k = 1; k <= 2 * i - 1; k++) {
       printf("*");
  }
      printf("\n");
   }
    return 0;


}
```



```
Enter the number of rows: 6
     *
    ***
   *****
  *******
 *********
***********
 *********
  *******
   *****
    ***
     *
```

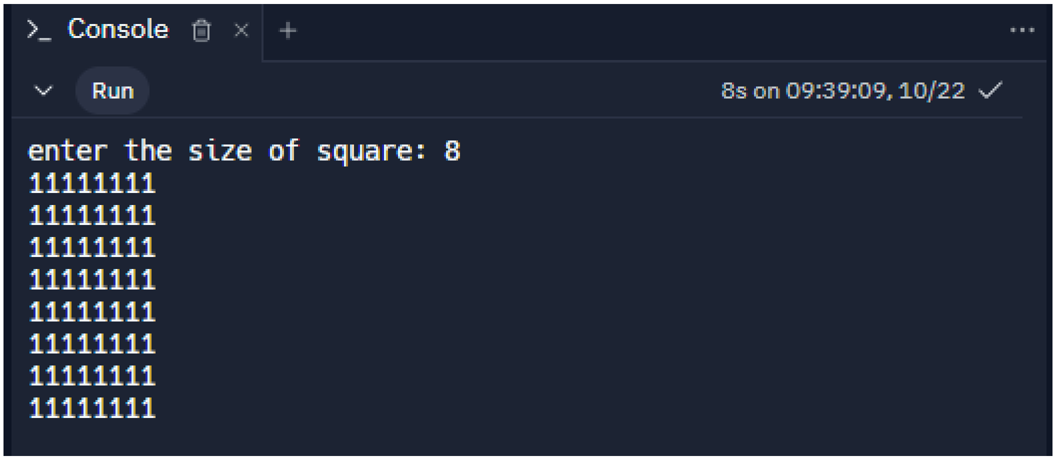
2. **Number pattern programs - Write a C program to print the given number patterns**

Square number patterns.

```
11111
11111
11111
11111
11111
```

```
#include <stdio.h>


int main (){

 int n,i,j;

  printf ("enter the size of square: ");

  scanf ("%d",&n);


 for (int i = 0; i < n; i++) {

  for (int j = 0; j < n; j++) {

   printf("1");

 }
```

```
    printf("\n");

 }

  return 0;

}
```



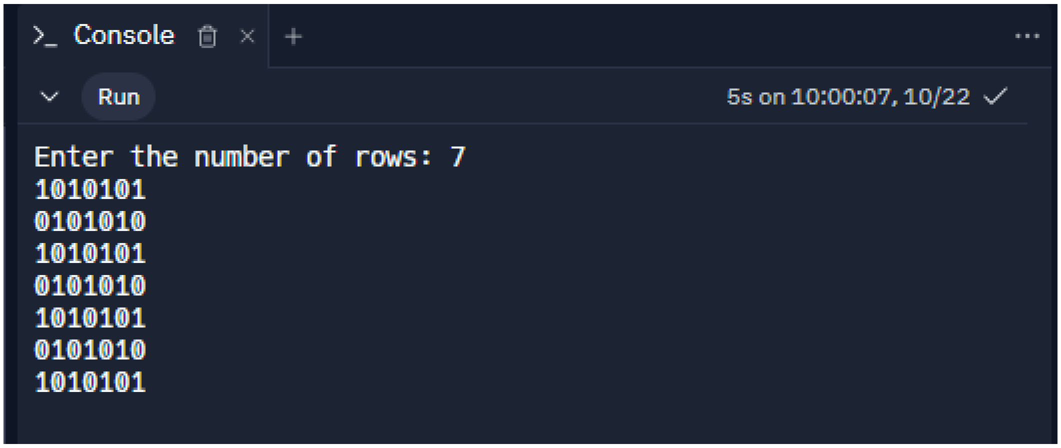
```
enter the size of square: 8
11111111
11111111
11111111
11111111
11111111
11111111
11111111
11111111
```

Number pattern 1.

```
11111
00000
11111
00000
11111
```

```
#include<stdio.h>

int main()

{

  int n;
```

```
    printf("Enter the number of rows: ");
    scanf("%d", &n);

    for (int i = 1; i <= n; i++) {
     for (int j = 1; j <= n; j++) {
    if ((i + j) % 2 == 0)
  {
       printf("1");
   }
      else
    {
      printf("0");
     }
   }
  printf("\n");
  }
 return 0;
}
```

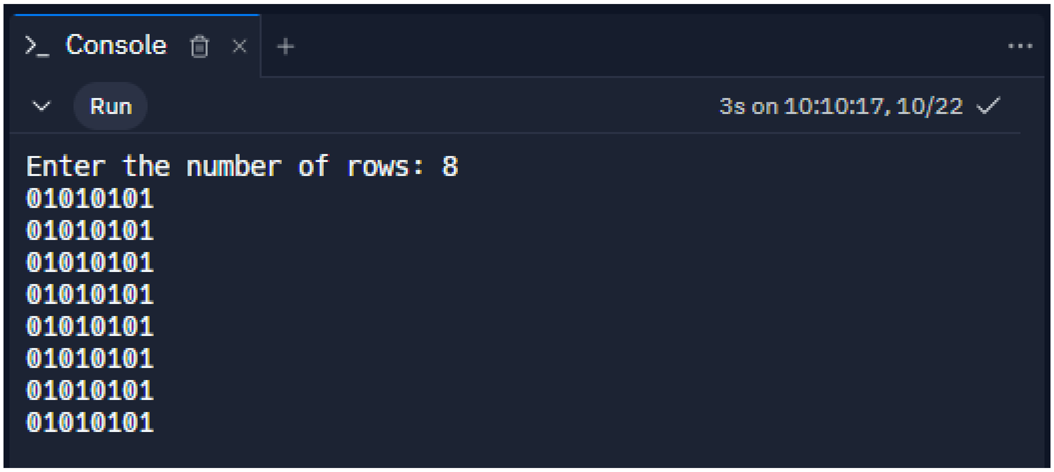


```
Enter the number of rows: 7
1010101
0101010
1010101
0101010
1010101
0101010
1010101
```

Number pattern 2.

```
01010
01010
01010
01010
01010
```

```
#include <stdio.h>


 int main() {

  int n;


   printf("Enter the number of rows: ");

   scanf("%d", &n);

   for (int i = 1; i <= n; i++) {

    for (int j = 1; j <= n; j++) {
      if (j % 2 == 0) {

       printf("1");

      } else

 {

      printf("0");

   }

  }

     printf("\n");

}

  return 0;

}
```



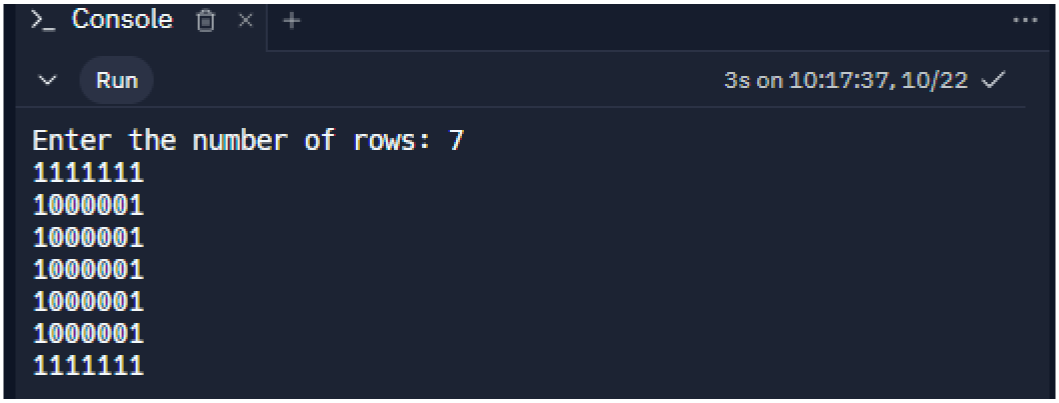
```
Enter the number of rows: 8
01010101
01010101
01010101
01010101
01010101
01010101
01010101
01010101
```

Number pattern 3.

```
11111
 10001
```

```
       10001
       10001
       11111
```

```c
#include <stdio.h>

int main() {
    int n;

    printf("Enter the number of rows: ");
    scanf("%d", &n);

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (j == 1 || j == n || i == 1 || i == n) {
                printf("1");
            } else {
                printf("0");
            }
        }
        printf("\n");
    }
  return 0;
}
```



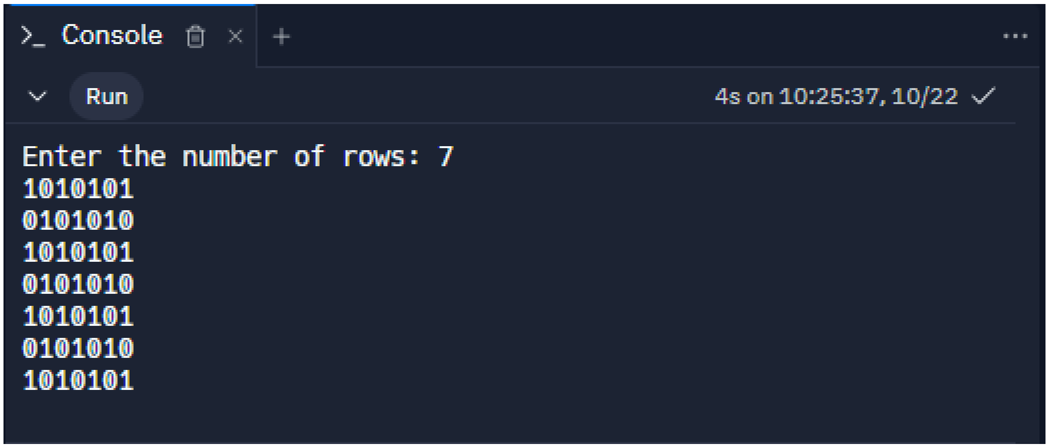
```
Enter the number of rows: 7
1111111
1000001
1000001
1000001
1000001
1000001
1111111
```

Number pattern 4.

```
            10101
             01010
             10101
             01010
             10101
```

```
#include <stdio.h>


int main() {
   int n;
    printf("Enter the number of rows: ");
    scanf("%d", &n);


   for (int i = 1; i <= n; i++) {
     if (i % 2 == 1) {
```

```
   for (int j = 1; j <= n; j++) {
    if (j % 2 == 1) {
      printf("1");
     }
else
  {
   printf("0");
    }
   }
   }
  else
  {
   for (int j = 1; j <= n; j++) {
    if (j % 2 == 1)
  {
    printf("0");
   } else
  {
   printf("1");

  }

  }

  }

   printf("\n");

  }

  return 0;

}
```

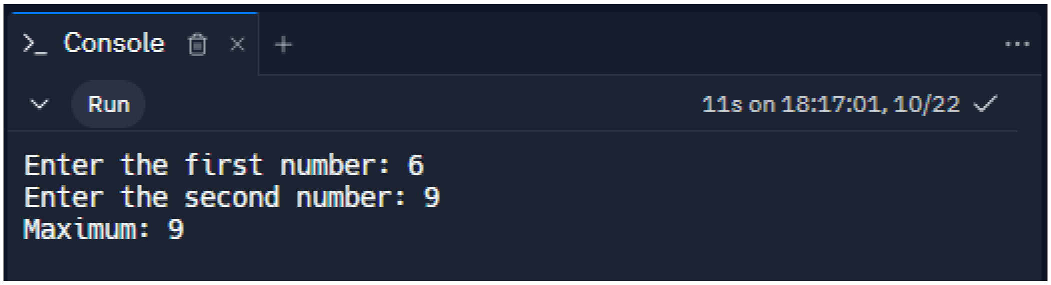


```
Enter the number of rows: 7
1010101
0101010
1010101
0101010
1010101
0101010
1010101
```

If..... Else Exercises.

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

```
#include <stdio.h>


int main() {
    int num1, num2;

    printf("Enter the first number: ");
    scanf("%d", &num1);

    printf("Enter the second number: ");
    scanf("%d", &num2);
    if (num1 > num2) {
        printf("Maximum: %d\n", num1);
    } else if (num2 > num1) {
        printf("Maximum: %d\n", num2);
    } else {
        printf("Both numbers are equal.\n");
    }
    return 0;
}
```

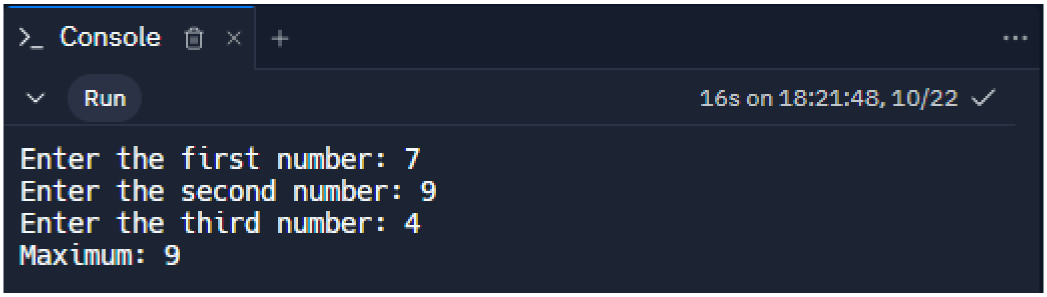


```
Enter the first number: 6
Enter the second number: 9
Maximum: 9
```

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

```
#include <stdio.h>



int main() {

    int num1, num2, num3;
```

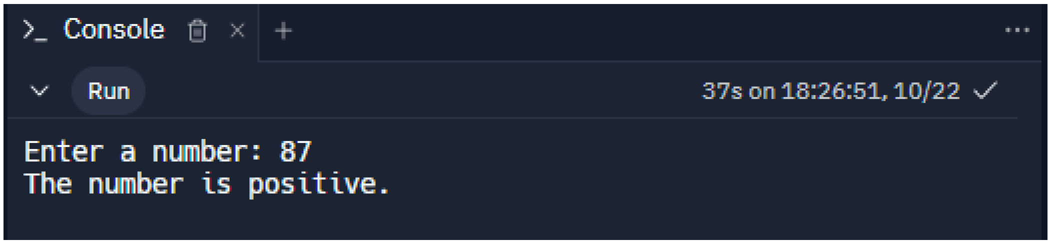
```
printf("Enter the first number: ");
scanf("%d", &num1);

printf("Enter the second number: ");
scanf("%d", &num2);

printf("Enter the third number: ");
scanf("%d", &num3);
if (num1 >= num2) {
    if (num1 >= num3) {

        printf("Maximum: %d\n", num1);
    } else {
        printf("Maximum: %d\n", num3);
  }
 } else {
    if (num2 >= num3) {
            printf("Maximum: %d\n", num2);
        } else {
         printf("Maximum: %d\n", num3);
       }
    }
    return 0;
}
```



```
Enter the first number: 7
Enter the second number: 9
Enter the third number: 4
Maximum: 9
```

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

```
#include <stdio.h>
int main() {
    int num;


    printf("Enter a number: ");
    scanf("%d", &num);


    if (num > 0) {
        printf("The number is positive.\n");
    } else if (num < 0) {
        printf("The number is negative.\n");
    } else {
        printf("The number is zero.\n");
    }
    return 0;
}
```

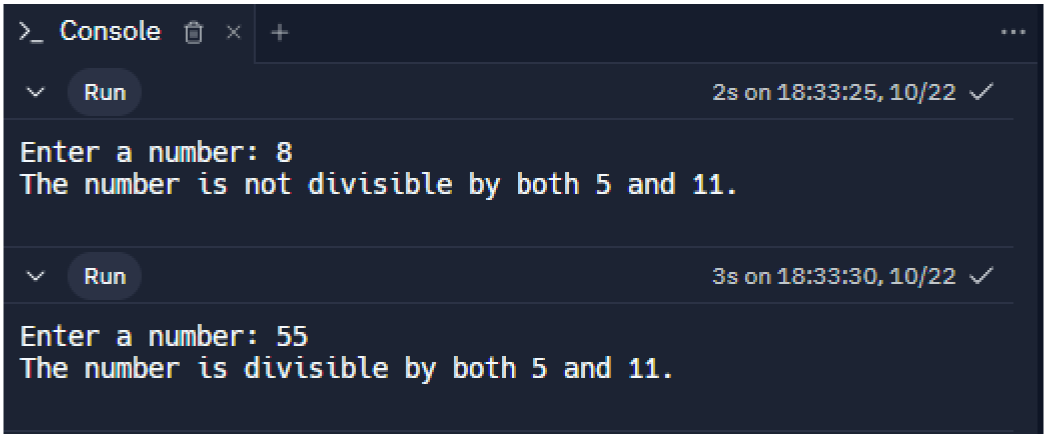


```
Enter a number: 87
The number is positive.
```

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

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

```
int main() {

    int num;

    printf("Enter a number: ");

    scanf("%d", &num);

    if (num % 5 == 0 && num % 11 == 0) {

        printf("The number is divisible by both 5 and
11.\n");

    } else {
        printf("The number is not divisible by both 5 and 11.\n");

    }

    return 0;

}
```



```
Enter a number: 8
The number is not divisible by both 5 and 11.
```

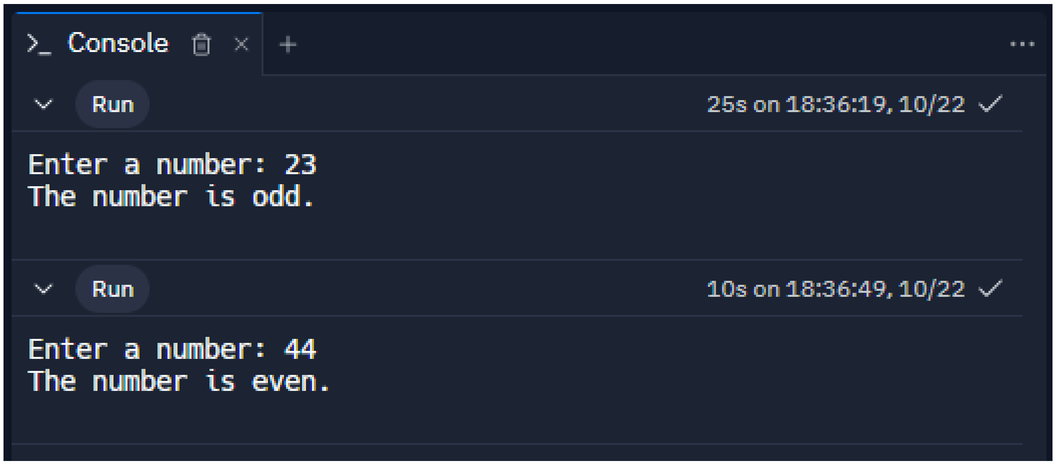
```
Enter a number: 55
The number is divisible by both 5 and 11.
```

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

```
#include <stdio.h>



int main() {
```

```
int num;
printf("Enter a number: ");
scanf("%d", &num);
if (num % 2 == 0) {
    printf("The number is even.\n");
} else {
    printf("The number is odd.\n");
}
```

```
    return 0;

}
```

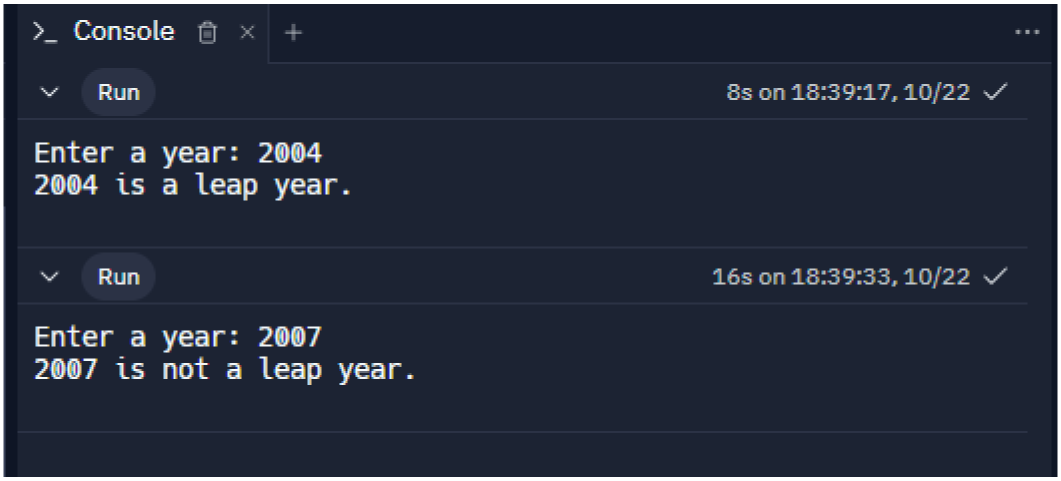


```
Enter a number: 23
The number is odd.
```

```
Enter a number: 44
The number is even.
```

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

```
#include <stdio.h>

int main() {
    int year;

    printf("Enter a year: ");

    scanf("%d", &year);

    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

        printf("%d is a leap year.\n", year);

    } else {

        printf("%d is not a leap year.\n", year);

    }
    return 0;

}
```



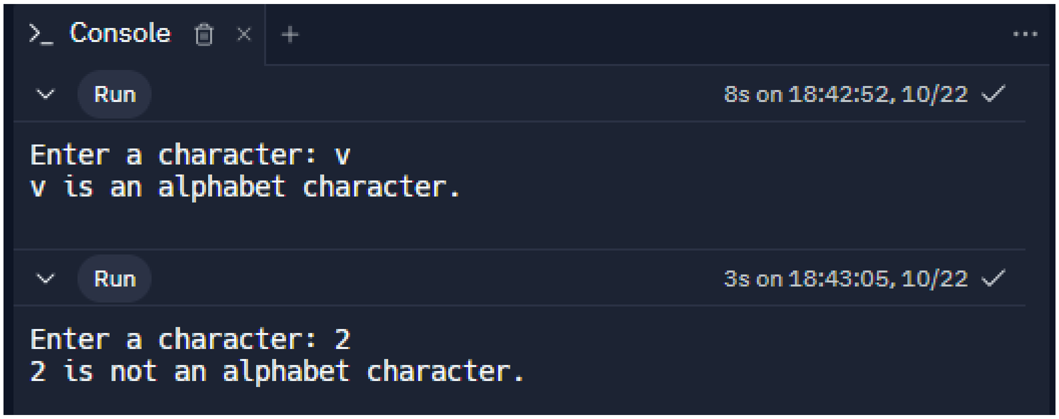
```
Enter a year: 2004
2004 is a leap year.
```

```
Enter a year: 2007
2007 is not a leap year.
```

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

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



int main() {
    char character;

    printf("Enter a character: ");

    scanf(" %c", &character);

    if (isalpha(character)) {

        printf("%c is an alphabet character.\n",
  character);

    } else {

        printf("%c is not an alphabet character.\n",
  character);
}
return 0;
```

```
}
```

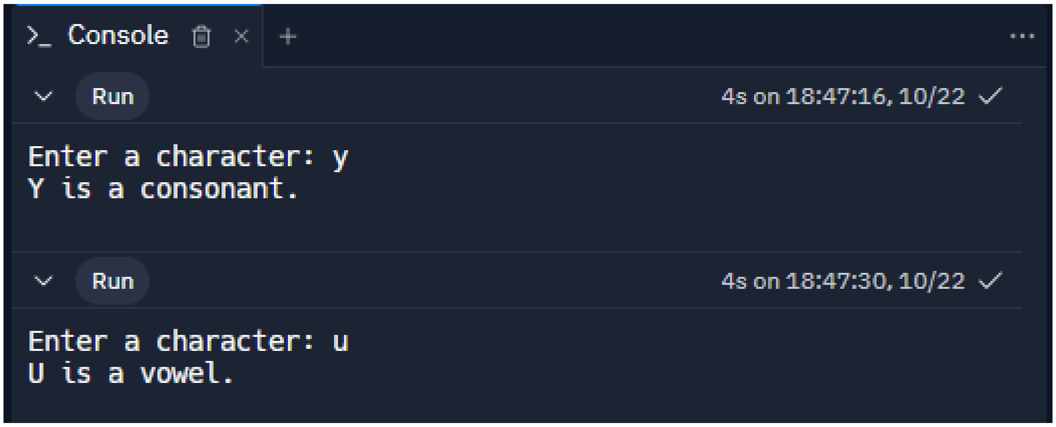


```
Enter a character: v
v is an alphabet character.
```

```
Enter a character: 2
2 is not an alphabet character.
```

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

```
#include <stdio.h>



#include <ctype.h>

int main() {



    char character;

    printf("Enter a character: ");

    scanf(" %c", &character);
    character = toupper(character);
    if (isalpha(character)) {

        if (character == 'A' || character == 'E' || character ==
'I' || character == 'O' || character == 'U') {

            printf("%c is a vowel.\n", character);
        } else {
            printf("%c is a consonant.\n", character);
        }
    } else {

        printf("Invalid input. Please enter an alphabet
character.\n");

    }    return 0;
}
```



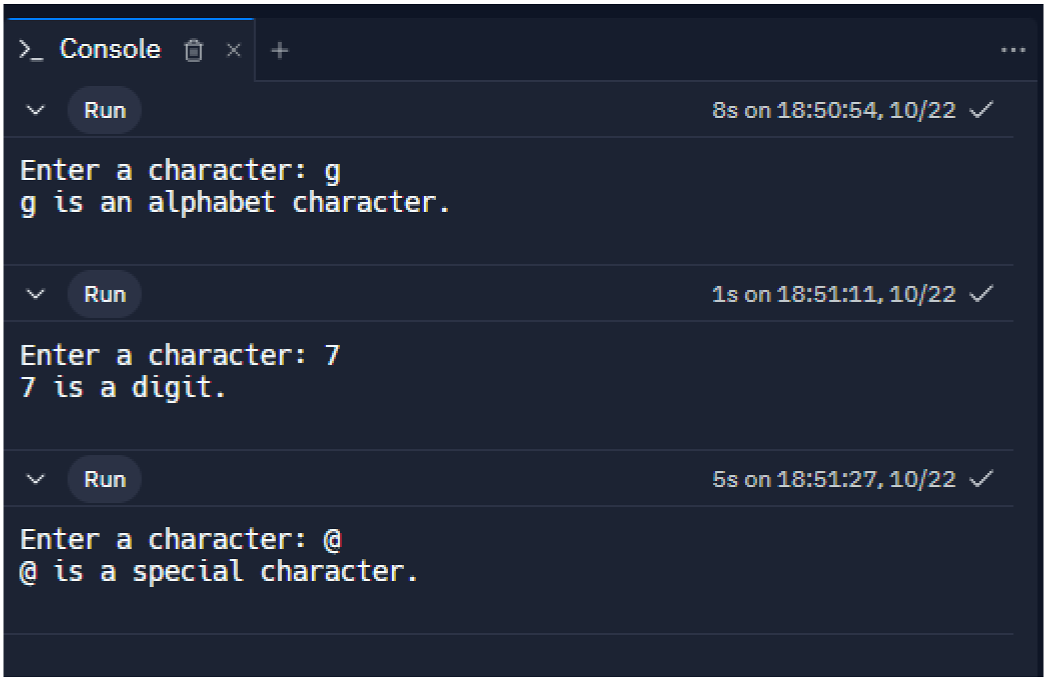
```
Enter a character: y
Y is a consonant.
```

```
Enter a character: u
U is a vowel.
```

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

```
#include <stdio.h>

#include <ctype.h>


int main() {

   char character;


   printf("Enter a character: ");

   scanf(" %c", &character);


   if (isalpha(character)) {

      printf("%c is an alphabet character.\n", character);
```

```
    } else if (isdigit(character)) {
        printf("%c is a digit.\n", character);
    } else {
        printf("%c is a special character.\n", character);
    }
    return 0;
}
```



```
Enter a character: g
g is an alphabet character.
```

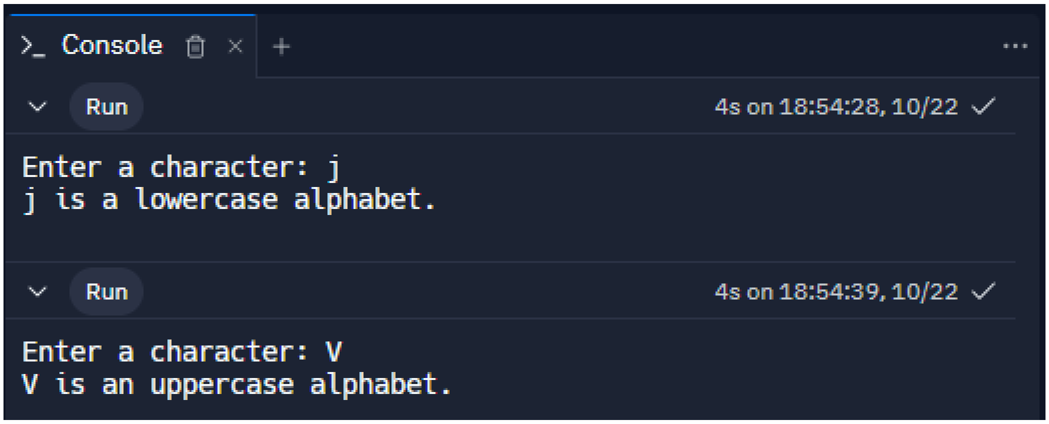
```
Enter a character: 7
7 is a digit.
```

```
Enter a character: @
@ is a special character.
```

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

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

```
#include <ctype.h>


int main() {
    char character;


    printf("Enter a character: ");
    scanf(" %c", &character);
    if (isupper(character)) {
        printf("%c is an uppercase alphabet.\n", character);


    } else if (islower(character)) {
        printf("%c is a lowercase alphabet.\n", character);
    } else {
        printf("%c is not an alphabet character.\n", character);


    }
```



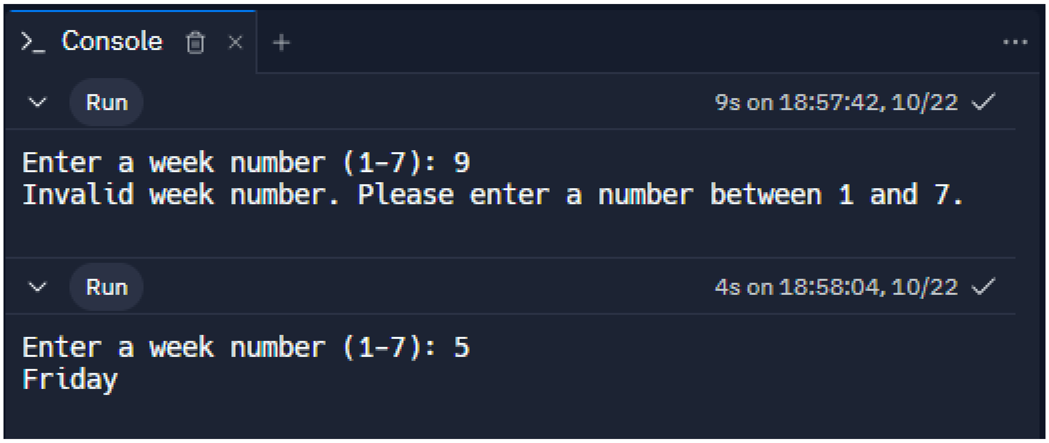
```
Enter a character: j
j is a lowercase alphabet.
```

```
Enter a character: V
V is an uppercase alphabet.
```

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

```
#include <stdio.h>



int main() {

    int weekNumber;

    printf("Enter a week number (1-7): ");

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

```
switch (weekNumber) {

    case 1:
        printf("Monday\n");
        break;

    case 2:
        printf("Tuesday\n");
        break;

    case 3:
        printf("Wednesday\n");
        break;

    case 4:
        printf("Thursday\n");
        break;
        case 5:
            printf("Friday\n");
            break;

        case 6:
            printf("Saturday\n");
            break;

        case 7:
            printf("Sunday\n");


            break;
        default:
            printf("Invalid week number. Please enter a number between 1 and 7.\n");


    }
    return 0;
}
```

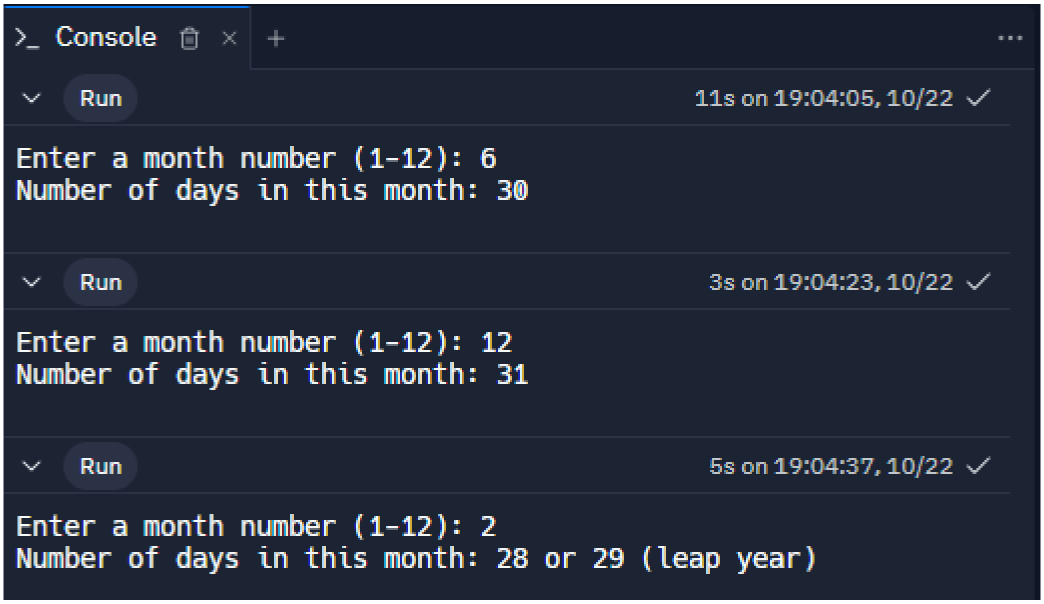


```
Enter a week number (1-7): 9
Invalid week number. Please enter a number between 1 and 7.
```

```
Enter a week number (1-7): 5
Friday
```

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

```
#include <stdio.h>




int main() {

    int monthNumber;


    printf("Enter a month number (1-12): ");

    scanf("%d", &monthNumber);
    switch (monthNumber) {
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            printf("Number of days in this month: 31\n");
            break;
        case 4: case 6: case 9: case 11:
            printf("Number of days in this month: 30\n");
            break;
        case 2:
            printf("Number of days in this month: 28 or 29 (leap year)\n");

            break;

        default:
            printf("Invalid month number. Please enter a number between 1 and 12.\n");
    }
    return 0;
}
```



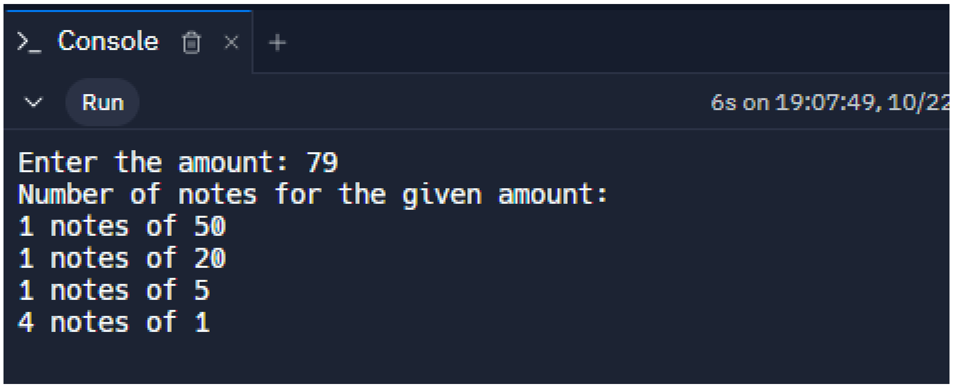
```
Enter a month number (1-12): 6
Number of days in this month: 30
```

```
Enter a month number (1-12): 12
Number of days in this month: 31
```

```
Enter a month number (1-12): 2
Number of days in this month: 28 or 29 (leap year)
```

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

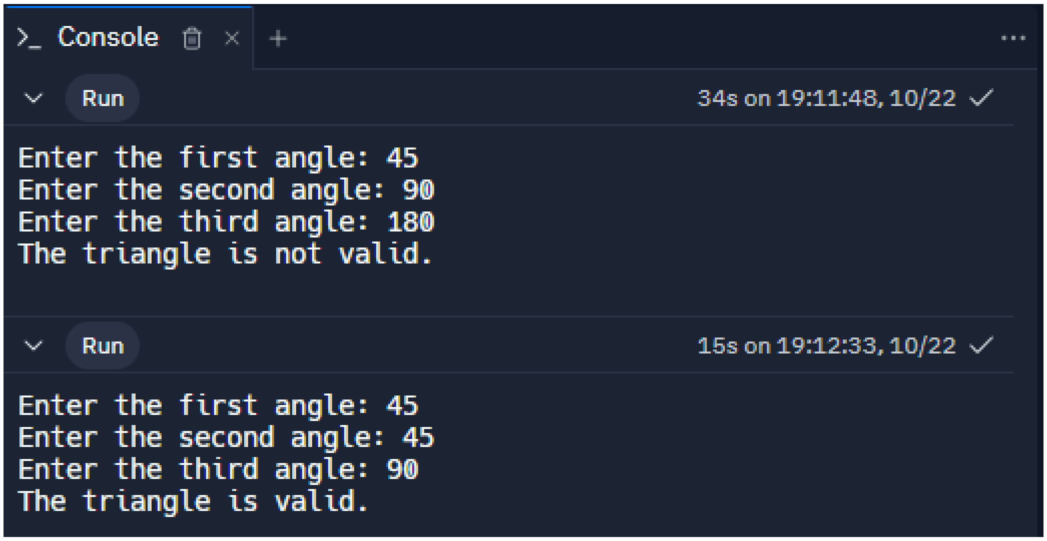
```
#include <stdio.h>


int main() {
    int amount;
    int notes[] = {2000, 500, 200, 100, 50, 20, 10, 5, 1};
    int count[9] = {0};


    printf("Enter the amount: ");
    scanf("%d", &amount);


    if (amount <= 0) {


        printf("Invalid amount. Please enter a positive
amount.\n");


    } else {
        printf("Number of notes for the given amount:\n");


        for (int i = 0; i < 9; i++) {


            if (amount >= notes[i]) {

                count[i] = amount / notes[i];


                amount = amount % notes[i];

                printf("%d notes of %d\n", count[i], notes[i]);

            }

        }

    }

    return 0;

}
```



```
Enter the amount: 79
Number of notes for the given amount:
1 notes of 50
1 notes of 20
1 notes of 5
4 notes of 1
```

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

```
#include <stdio.h>
int main() {

    float angle1, angle2, angle3;

    printf("Enter the first angle: ");
    scanf("%f", &angle1);

    printf("Enter the second angle: ");
    scanf("%f", &angle2);

    printf("Enter the third angle: ");
    scanf("%f", &angle3);
    if (angle1 + angle2 + angle3 == 180) {

        printf("The triangle is valid.\n");
    } else {
   printf("The triangle is not valid.\n");
    }
    return 0;
}
```



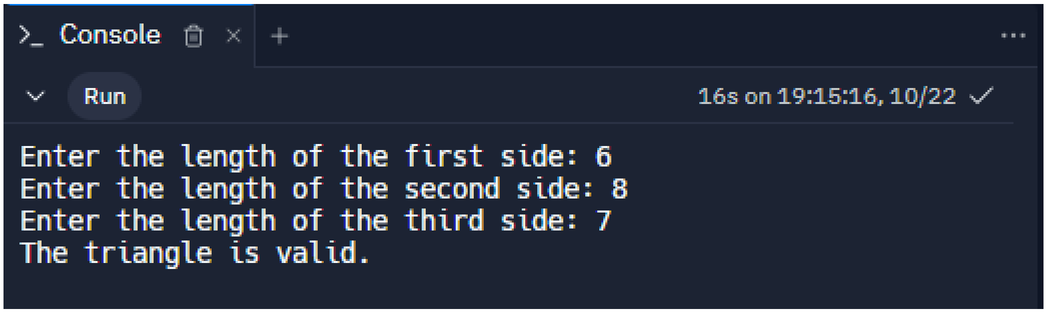
```
Enter the first angle: 45
Enter the second angle: 90
Enter the third angle: 180
The triangle is not valid.
```

```
Enter the first angle: 45
Enter the second angle: 45
Enter the third angle: 90
The triangle is valid.
```

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

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

```
int main() {

    float side1, side2, side3;

    printf("Enter the length of the first side: ");

    scanf("%f", &side1);

    printf("Enter the length of the second side: ");
    scanf("%f", &side2);

    printf("Enter the length of the third side: ");
    scanf("%f", &side3);

    if ((side1 + side2 > side3) && (side1 + side3 >
side2) && (side2 + side3 > side1)) {
        printf("The triangle is valid.\n");
    } else {


        printf("The triangle is not valid.\n");
    }
    return 0;
  }
```

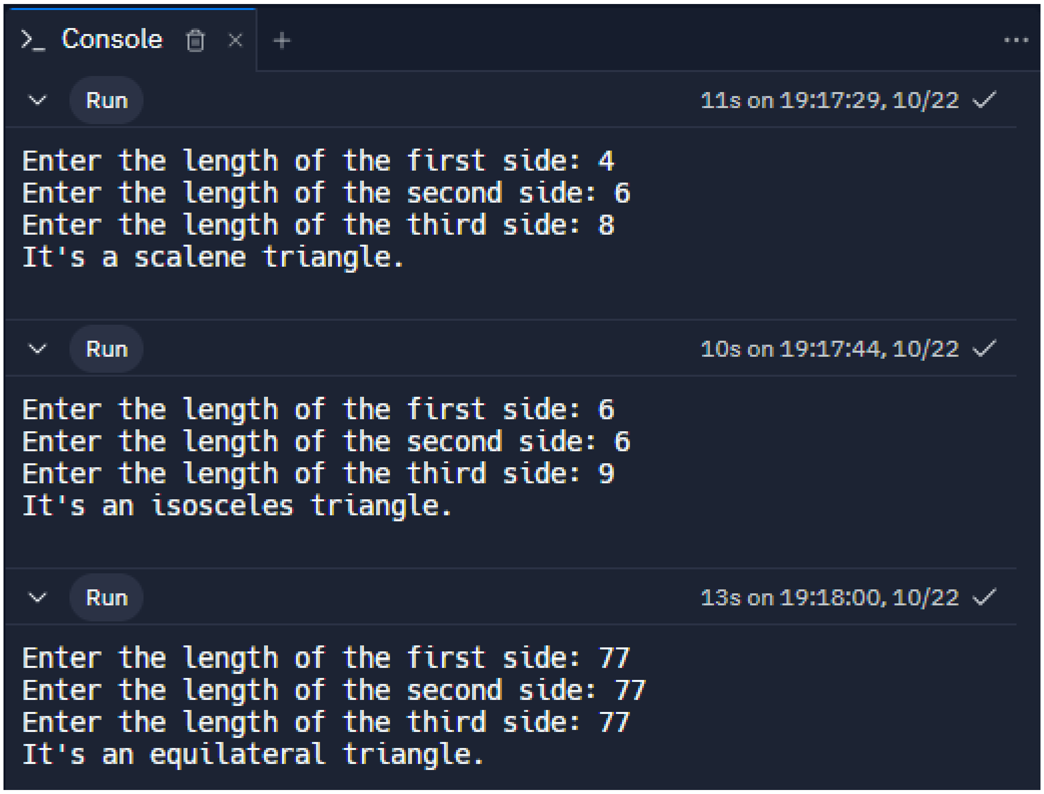


```
Enter the length of the first side: 6
Enter the length of the second side: 8
Enter the length of the third side: 7
The triangle is valid.
```

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

```
#include <stdio.h>


int main() {
    float side1, side2, side3;

    printf("Enter the length of the first side: ");
    scanf("%f", &side1);

    printf("Enter the length of the second side: ");
    scanf("%f", &side2);

    printf("Enter the length of the third side: ");
    scanf("%f", &side3);

    if ((side1 + side2 > side3) && (side1 + side3 > side2)
&& (side2 + side3 > side1)) {
        if (side1 == side2 && side2 == side3) {
            printf("It's an equilateral triangle.\n");

        } else if (side1 == side2 || side1 == side3 || side2
== side3) {
```

```
            printf("It's an isosceles triangle.\n");
        } else {
            printf("It's a scalene triangle.\n");
        }
    } else {
        printf("It's not a valid triangle.\n");
    }
    return 0;
}
```



```
Enter the length of the first side: 4
Enter the length of the second side: 6
Enter the length of the third side: 8
It's a scalene triangle.
```

```
Enter the length of the first side: 6
Enter the length of the second side: 6
Enter the length of the third side: 9
It's an isosceles triangle.
```

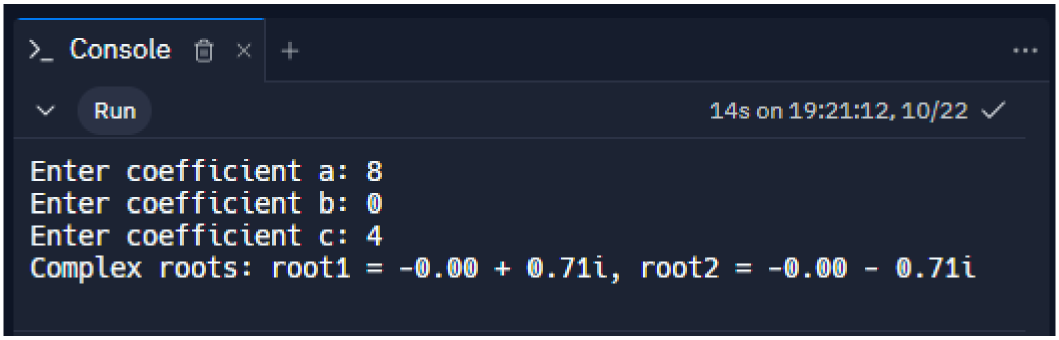
```
Enter the length of the first side: 77
Enter the length of the second side: 77
Enter the length of the third side: 77
It's an equilateral triangle.
```

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

```
#include <math.h>



int main() {



    double a, b, c;

    double discriminant, root1, root2;
    printf("Enter coefficient a: ");
    scanf("%lf", &a);

    printf("Enter coefficient b: ");
    scanf("%lf", &b);

    printf("Enter coefficient c: ");
    scanf("%lf", &c);

    discriminant = b * b - 4 * a * c;

    if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);

        printf("Two distinct real roots: root1 = %.2lf,
root2 = %.2lf\n", root1, root2);
```

```
    } else if (discriminant == 0) {

        root1 = root2 = -b / (2 * a);

        printf("Two equal real roots: root1 = root2 = %.2lf\n", root1);
    } else {
        double realPart = -b / (2 * a);
        double imaginaryPart = sqrt(-discriminant) / (2 * a);

        printf("Complex roots: root1 = %.2lf + %.2lfi, root2 = %.2lf - %.2lfi\n", realPart, imaginaryPart, realPart, imaginaryPart);

    }
 return 0;
```

```
}
```

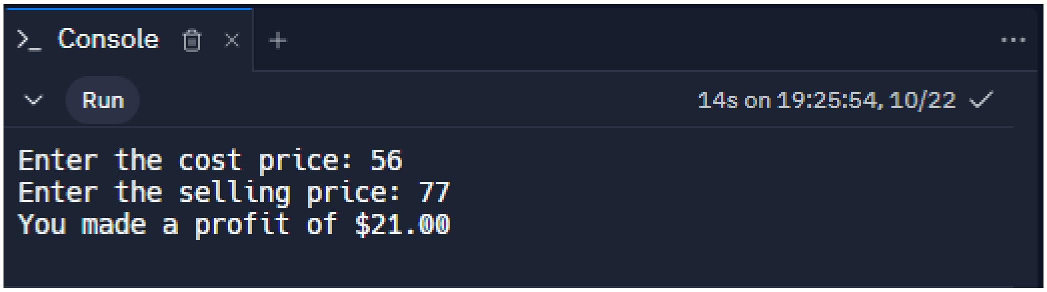


```
Enter coefficient a: 8
Enter coefficient b: 0
Enter coefficient c: 4
Complex roots: root1 = -0.00 + 0.71i, root2 = -0.00 - 0.71i
```

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

```
#include <stdio.h>

int main() {
   float costPrice, sellingPrice, profitOrLoss;
   printf("Enter the cost price: ");
   scanf("%f", &costPrice);

   printf("Enter the selling price: ");
   scanf("%f", &sellingPrice);
   profitOrLoss = sellingPrice - costPrice;

   if (profitOrLoss > 0) {
      printf("You made a profit of $%.2f\n", profitOrLoss);
   } else if (profitOrLoss < 0) {
      printf("You incurred a loss of $%.2f\n", -profitOrLoss);
```

```
    } else {
        printf("You broke even; no profit or loss.\n");
    }

    return 0;
}
```



```
Enter the cost price: 56
Enter the selling price: 77
You made a profit of $21.00
```

19. Write a C program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer. Calculate percentage and grade according to following:

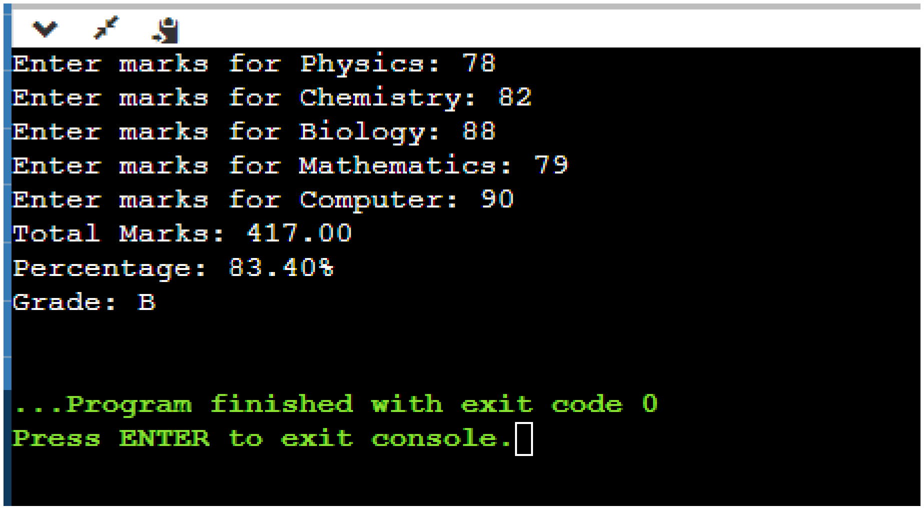
Percentage >= 90% : Grade A
Percentage >= 80% : Grade B
Percentage >= 70% : Grade C
Percentage >= 60% : Grade D
Percentage >= 40% : Grade E
Percentage < 40% : Grade F

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

```
int main() {

    float physics, chemistry, biology, mathematics, computer;

    float totalMarks, percentage;

    char grade;

    printf("Enter marks for Physics: ");

    scanf("%f", &physics);

    printf("Enter marks for Chemistry: ");

    scanf("%f", &chemistry);

    printf("Enter marks for Biology: ");

    scanf("%f", &biology);

    printf("Enter marks for Mathematics: ");

    scanf("%f", &mathematics);

    printf("Enter marks for Computer: ");

    scanf("%f", &computer);

    totalMarks = physics + chemistry + biology + mathematics
+ computer;
```

```
percentage = (totalMarks / 500) * 100;
if (percentage >= 90) {
    grade = 'A';
} else if (percentage >= 80) {
    grade = 'B';
} else if (percentage >= 70) {
    grade = 'C';
} else if (percentage >= 60) {
    grade = 'D';
} else if (percentage >= 40) {
    grade = 'E';
} else {
    grade = 'F';
}
printf("Total Marks: %.2f\n", totalMarks);
```

```
    printf("Percentage: %.2f%%\n", percentage);

    printf("Grade: %c\n", grade);

     return 0;

}
```



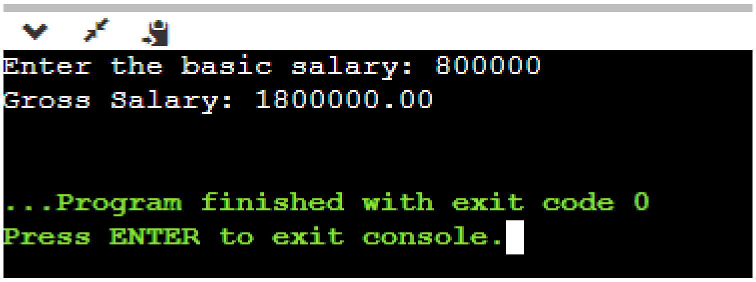
```
Enter marks for Physics: 78
Enter marks for Chemistry: 82
Enter marks for Biology: 88
Enter marks for Mathematics: 79
Enter marks for Computer: 90
Total Marks: 417.00
Percentage: 83.40%
Grade: B


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

20. 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%

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

```
int main() {

    float basicSalary, grossSalary;

    float hra, da;

    printf("Enter the basic salary: ");

    scanf("%f", &basicSalary);

    if (basicSalary <= 10000) {

        hra = 0.2 * basicSalary;

        da = 0.8 * basicSalary;

    } else if (basicSalary <= 20000) {

        hra = 0.25 * basicSalary;

        da = 0.9 * basicSalary;

    } else {
        hra = 0.3 * basicSalary;

        da = 0.95 * basicSalary;

    }

    grossSalary = basicSalary + hra + da;

    printf("Gross Salary: %.2f\n", grossSalary);



    return 0;

}
```



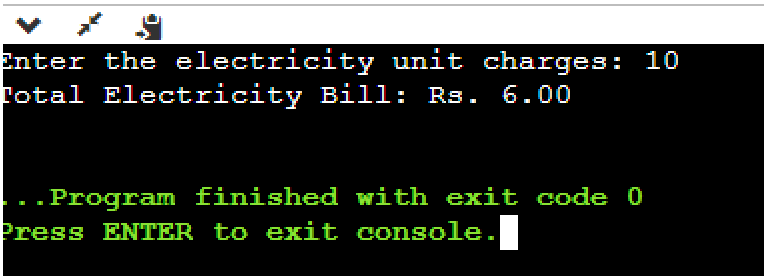
```
Enter the basic salary: 800000
Gross Salary: 1800000.00


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

21. 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

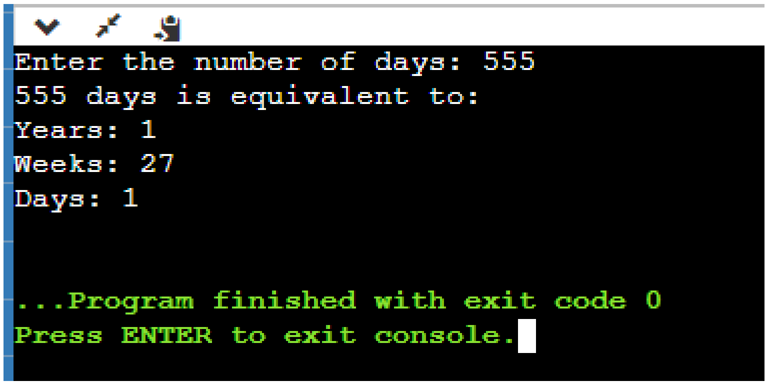
```
#include <stdio.h>



int main() {

    float unitCharges, totalBill;

    printf("Enter the electricity unit charges: ");

    scanf("%f", &unitCharges);



    if (unitCharges <= 50) {

        totalBill = unitCharges * 0.50;

    } else if (unitCharges <= 150) {

        totalBill = (50 * 0.50) + ((unitCharges - 50) * 0.75);

    } else if (unitCharges <= 250) {
        totalBill = (50 * 0.50) + (100 * 0.75) + ((unitCharges -
150) * 1.20);

    } else {

        totalBill = (50 * 0.50) + (100 * 0.75) + (100 * 1.20) +
((unitCharges - 250) * 1.50);

    }

    totalBill += 0.20 * totalBill;

    printf("Total Electricity Bill: Rs. %.2f\n", totalBill);



    return 0;

}
```



```
Enter the electricity unit charges: 10
Total Electricity Bill: Rs. 6.00


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

22. Write a C program to convert specified days into years, weeks and days .

```
#include <stdio.h>



int main() {

    int days, years, weeks, remainingDays;

    printf("Enter the number of days: ");

    scanf("%d", &days);

    years = days / 365;

    weeks = (days % 365) / 7;

    remainingDays = days - (years * 365) - (weeks * 7);

    printf("%d days is equivalent to:\n", days);

    printf("Years: %d\n", years);

    printf("Weeks: %d\n", weeks);

    printf("Days: %d\n", remainingDays);
    return 0;

}
```



```
Enter the number of days: 555
555 days is equivalent to:
Years: 1
Weeks: 27
Days: 1


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