**Lab file**

Introduction to c programming



Batch-2023-2027

Bca(ai&ds)

Submitted by - Submitted to -

Abhishek singh Rathore Mr. Rishi

(23160119) kumar

(Assistant Professor-CSIT GEU)

**INDEX**

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

**2. WAP to add two numbers.**

**3. WAP to find area of circle.**

**4. WAP to divide two numbers.**

**5. WAP to print ASCII value.**

**6. WAP to multiply floating point numbers.**

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

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

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

**10.Wap to find the area of rectangle.**

**11. WAP to find area of square**

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

**13. wap to find Area and Volume of Cube.**

**14. wap to find area and volume of cuboid.**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**72 Write a C program to find maximum between two numbers.**

**73.Write a C program to find maximum between three numbers.**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**100. Write a C program to calculate profit or loss.**

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

**102. Write a C program to input basic salary of an employee and calculate its Gross salary according**

**103. Write a C program to input electricity unit charges and calculate total electricity bill according**

**104. Write a C program to convert specified days into years, weeks and day.**

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

#include <stdio.h>

int main() {

int n;

// Input the value of n from the user

printf("Enter a positive integer: ");

scanf("%d", &n);

if (n < 1) {

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

} else {

// Print natural numbers in reverse order

printf("Natural numbers in reverse order from %d to 1:\n", n);

while (n >= 1) {

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

n--;

}

}

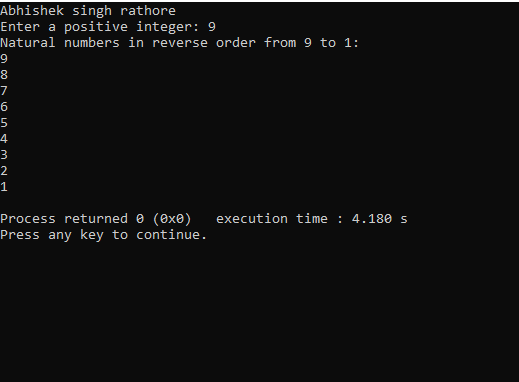
return 0;

}

}

Nnm

Nj



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

#include <stdio.h>

int main() {

char alphabet;

// Start from 'A' and loop until 'Z'

for (alphabet = 'A'; alphabet <= 'Z'; alphabet++) {

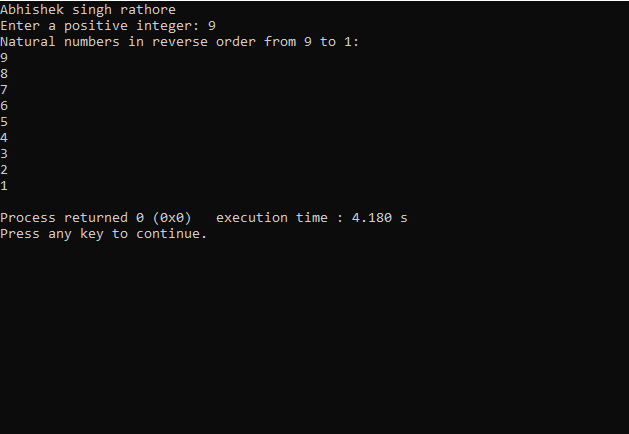
printf("%c ", alphabet);

}

printf("\n"); // Print a newline to end the line

return 0;

}



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

#include <stdio.h>

int main() {

printf(“abhishek singh rathore”);

int n, i;

// Ask the user for the value of n

printf("Enter a positive integer n: ");

scanf("%d", &n);

if (n <= 0) {

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

} else {

// Loop from 1 to n and print each number

printf("Natural numbers from 1 to %d are:\n", n);

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

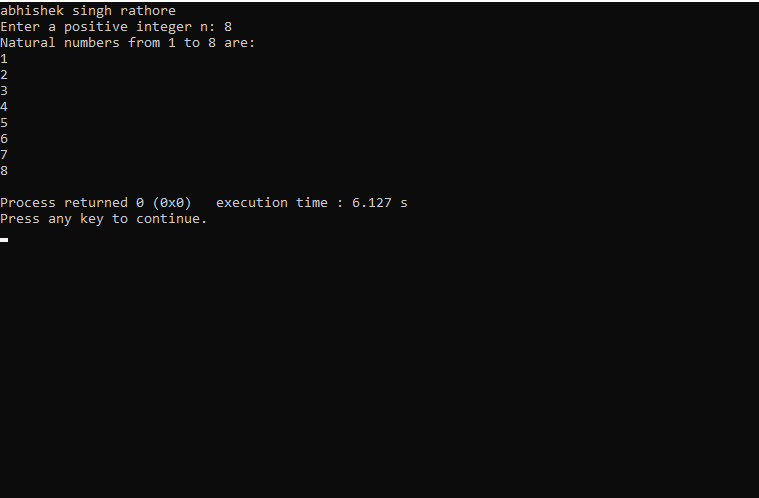
printf("%d\n", i);

}

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int i;

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

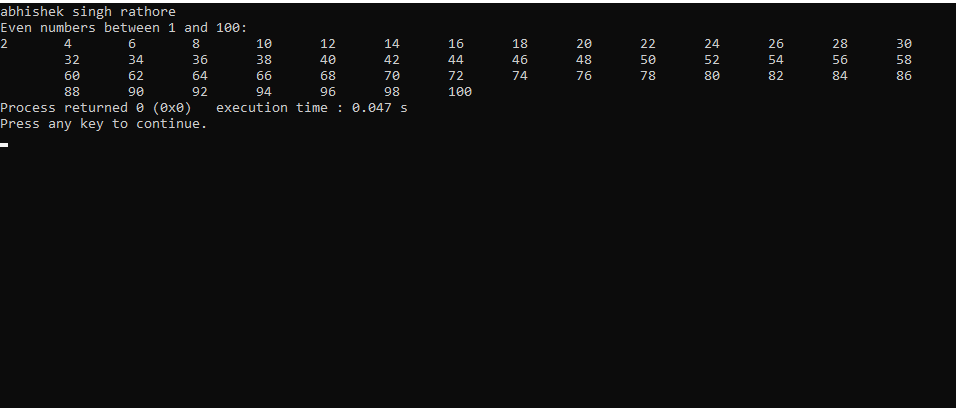
for (i = 2; i <= 100; i += 2) {

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

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int i;

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

for (i = 1; i <= 100; i++) {

if (i % 2 != 0) {

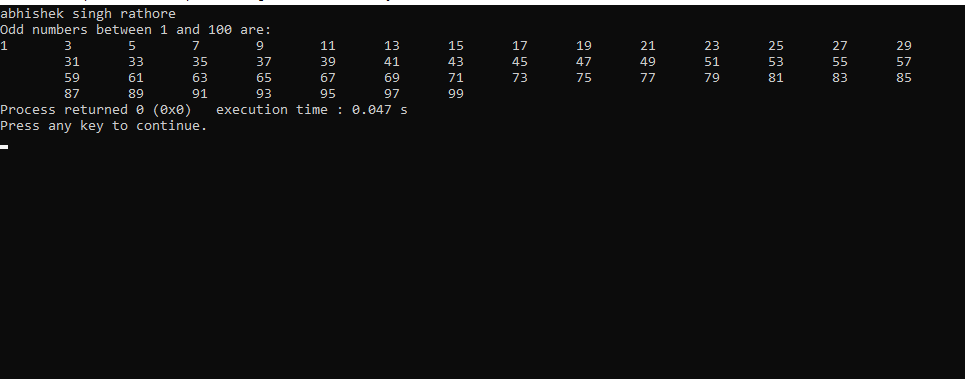
printf("%d\t", i);

}

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int n, sum = 0;

// Prompt the user to enter a positive integer

printf("Enter a positive integer (n): ");

scanf("%d", &n);

if (n <= 0) {

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

} else {

// Calculate the sum of natural numbers from 1 to n

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

sum += i;

}

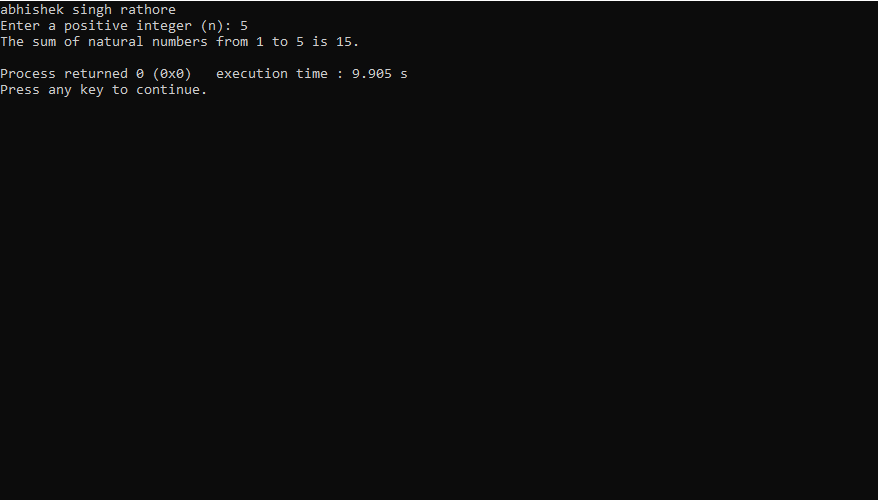
// Display the result

printf("The sum of natural numbers from 1 to %d is %d.\n", n, sum);

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int n, sum = 0;

// Input the value of n

printf("Enter a positive integer n: ");

scanf("%d", &n);

// Check if n is a positive integer

if (n <= 0) {

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

} else {

// Calculate the sum of even natural numbers

for (int i = 2; i <= n; i += 2) {

sum += i;

}

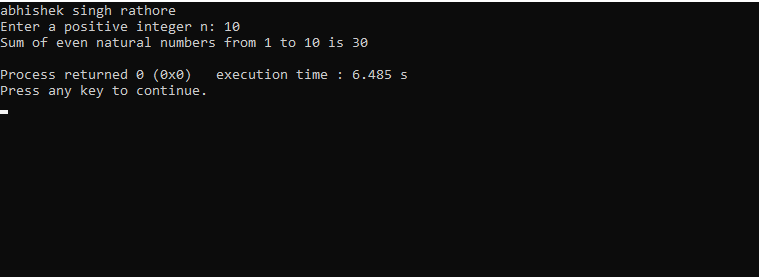
// Display the result

printf("Sum of even natural numbers from 1 to %d is %d\n", n, sum);

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int n, sum = 0;

printf("Enter a positive integer (n): ");

scanf("%d", &n);

if (n <= 0) {

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

} else {

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

if (i % 2 != 0) { // Check if the number is odd

sum += i;

}

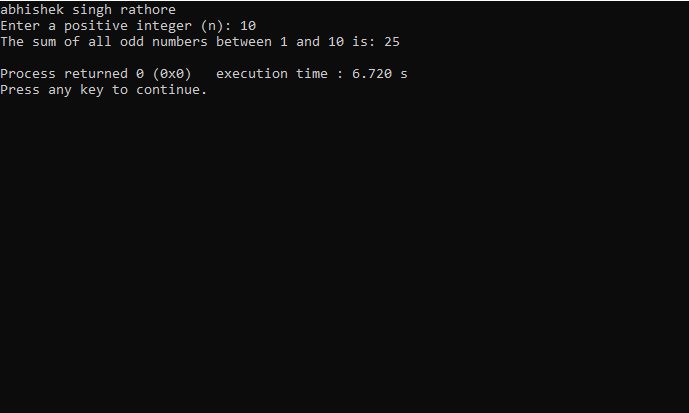
}

printf("The sum of all odd numbers between 1 and %d is: %d\n", n, sum);

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int number, i;

// Ask the user to enter a number

printf("Enter a number: ");

scanf("%d", &number);

// Print the multiplication table of the entered number

printf("Multiplication Table for %d:\n", number);

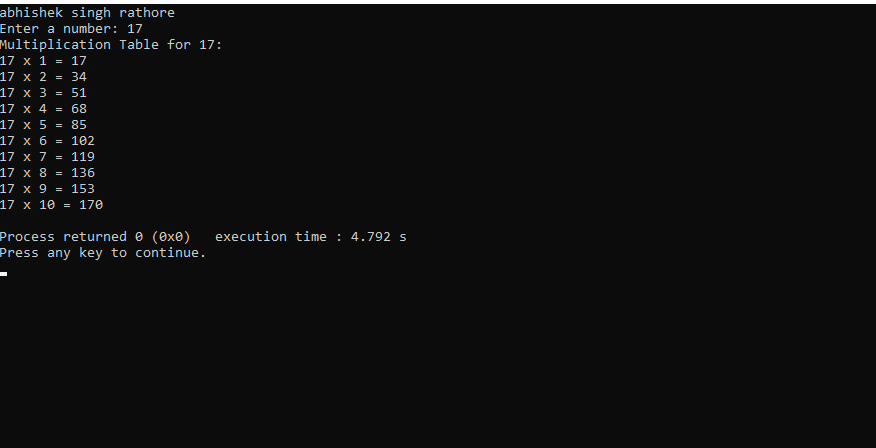
for (i = 1; i <= 10; i++) {

printf("%d x %d = %d\n", number, i, number \* i);

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int number, count = 0;

printf("Enter an integer: ");

scanf("%d", &number);

// Ensure the number is non-negative

if (number < 0) {

number = -number;

}

if (number == 0) {

count = 1; // If the number is 0, it has one digit

} else {

while (number > 0) {

number = number / 10; // Remove the last digit

count++; // Increment the digit count

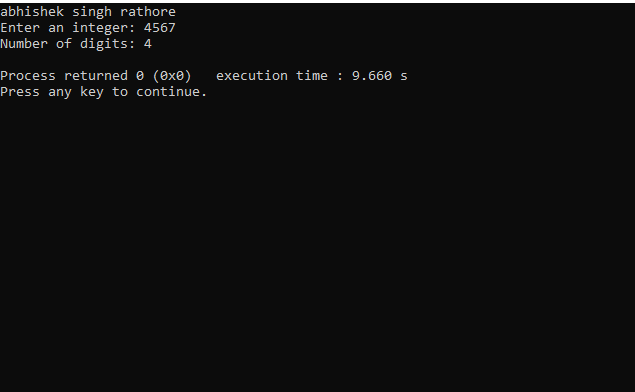
}

}

printf("Number of digits: %d\n", count);

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int num, firstDigit, lastDigit;

// Input a number from the user

printf("Enter an integer: ");

scanf("%d", &num);

// Find the last digit

lastDigit = num % 10;

// Find the first digit

while (num >= 10) {

num /= 10;

}

firstDigit = num;

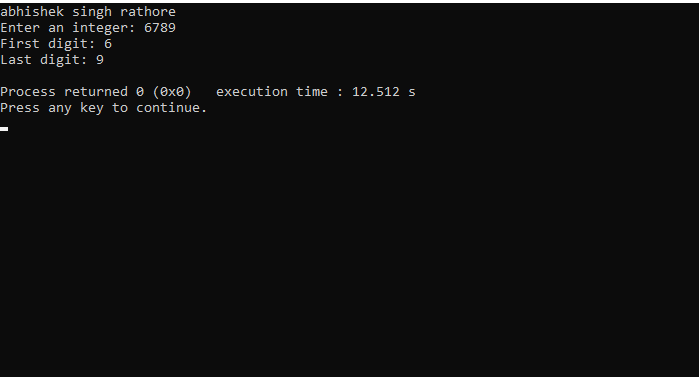
// Display the first and last digits

printf("First digit: %d\n", firstDigit);

printf("Last digit: %d\n", lastDigit);

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int num, firstDigit, lastDigit, sum;

// Input a number

printf("Enter a number: ");

scanf("%d", &num);

// Extract the last digit

lastDigit = num % 10;

// Find the first digit

while (num >= 10) {

num /= 10;

}

firstDigit = num;

// Calculate the sum

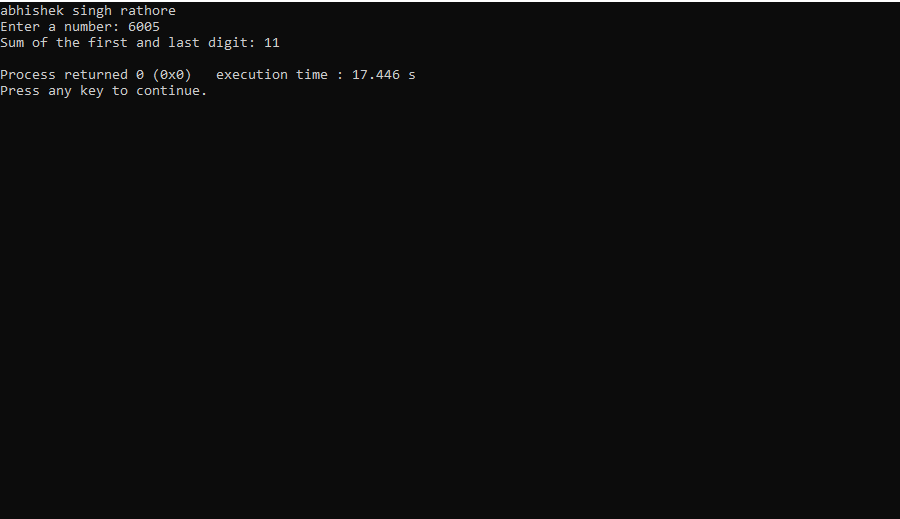
sum = firstDigit + lastDigit;

// Display the result

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

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int num, originalNum, firstDigit, lastDigit, reversedNum;

// Input the number

printf("Enter an integer: ");

scanf("%d", &num);

originalNum = num;

// Extract the last digit

lastDigit = num % 10;

// Find the number of digits in the given number

int numDigits = 0;

while (num != 0) {

num /= 10;

numDigits++;

}

// Reset num to the original number

num = originalNum;

// Extract the first digit

firstDigit = num / (int)pow(10, numDigits - 1);

// Remove the first and last digits

num = originalNum % (int)pow(10, numDigits - 1);

num = num / 10;

// Swap the first and last digits and restore the middle digits

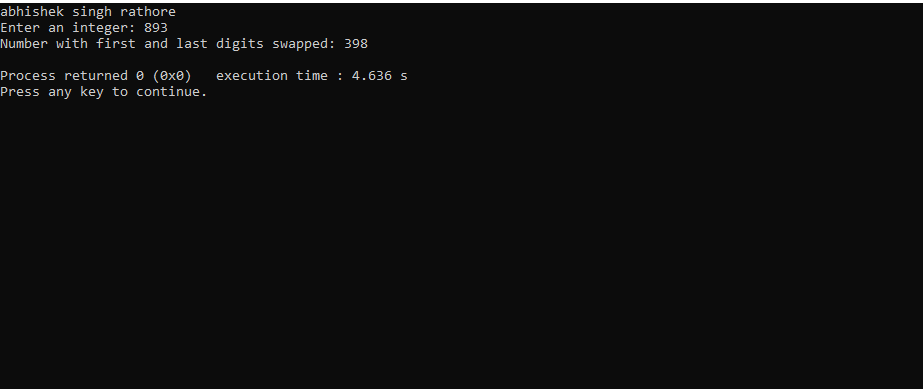
reversedNum = lastDigit \* (int)pow(10, numDigits - 1) + num \* 10 + firstDigit;

// Output the result

printf("Number with first and last digits swapped: %d\n", reversedNum);

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int num, digit, sum = 0;

// Input from the user

printf("Enter an integer: ");

scanf("%d", &num);

// Calculate the sum of digits

while (num != 0) {

digit = num % 10; // Get the last digit

sum += digit; // Add the digit to the sum

num /= 10; // Remove the last digit

}

// Display the result

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

return 0;

}

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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int number, product = 1;

// Input the number from the user

printf("Enter a number: ");

scanf("%d", &number);

// Ensure the number is non-negative

if (number < 0) {

printf("Please enter a non-negative number.\n");

return 1; // Exit with an error code

}

// Calculate the product of digits

while (number > 0) {

int digit = number % 10;

product \*= digit;

number /= 10;

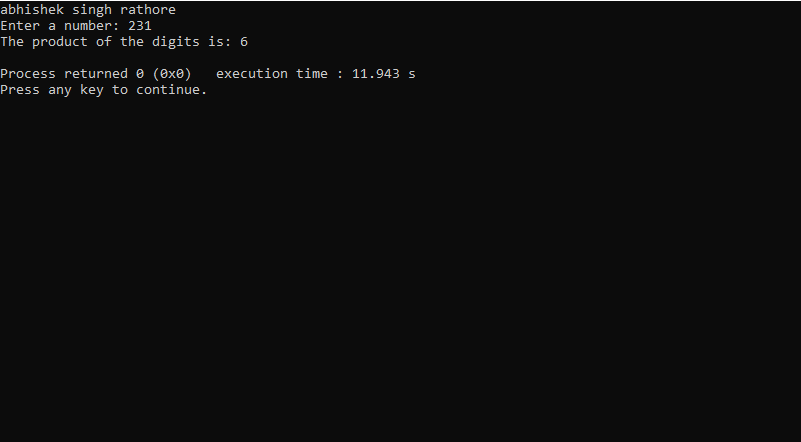
}

// Output the result

printf("The product of the digits is: %d\n", product);

return 0; // Exit with a success code

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathor\n");

int number, reverse = 0, remainder;

printf("Enter a number: ");

scanf("%d", &number);

while (number != 0) {

remainder = number % 10;

reverse = reverse \* 10 + remainder;

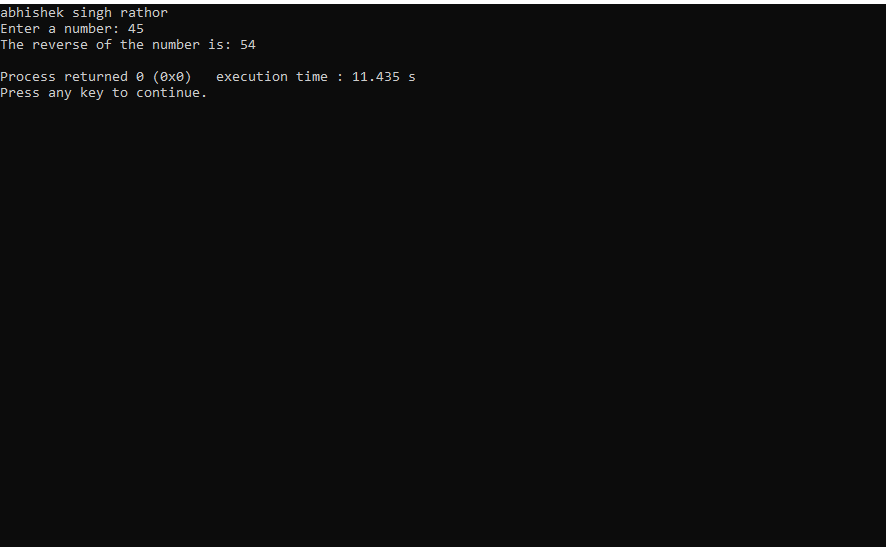
number = number / 10;

}

printf("The reverse of the number is: %d\n", reverse);

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int num, reversedNum = 0, originalNum, remainder;

printf("Enter an integer: ");

scanf("%d", &num);

originalNum = num;

// Reverse the number

while (num > 0) {

remainder = num % 10;

reversedNum = reversedNum \* 10 + remainder;

num /= 10;

}

// Check if the number is a palindrome

if (originalNum == reversedNum) {

printf("%d is a palindrome.\n", originalNum);

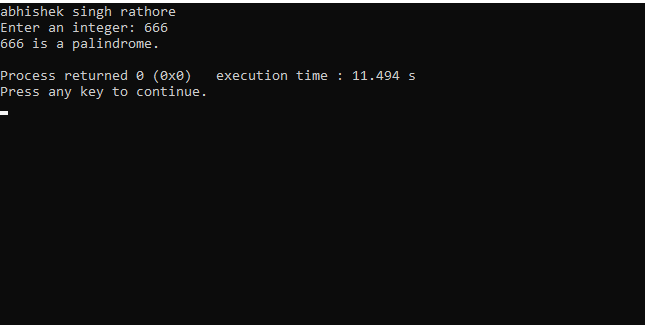
} else {

printf("%d is not a palindrome.\n", originalNum);

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int num;

printf("Enter an integer: ");

scanf("%d", &num);

// Initialize an array to store the frequency of each digit (0-9)

int digitCount[10] = {0};

// Process each digit in the integer

while (num != 0) {

int digit = num % 10; // Get the last digit

digitCount[digit]++; // Increment the count for that digit

num /= 10; // Remove the last digit

}

// Display the frequency of each digit

printf("Digit Frequency\n");

for (int i = 0; i < 10; i++) {

if (digitCount[i] > 0) {

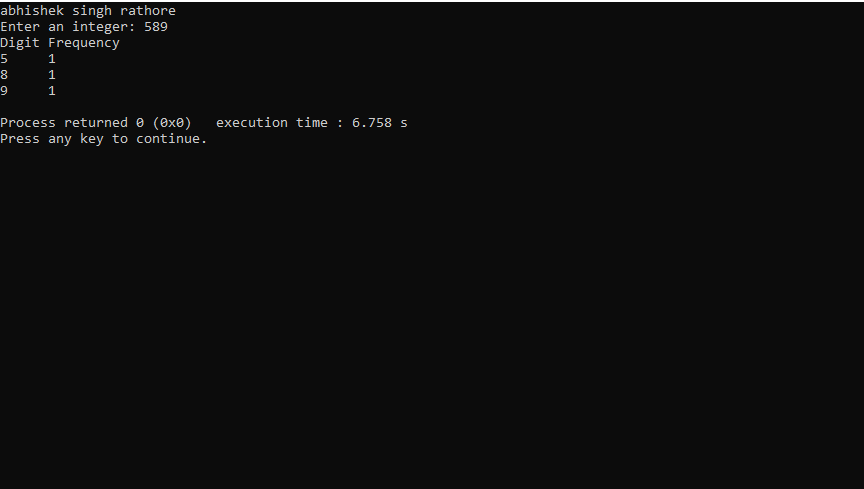
printf("%d %d\n", i, digitCount[i]);

}

}

return 0;

}



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

#include <stdio.h>

void printNumberInWords(int number);

int main() {

printf("abhishek singh rathore\n");

int number;

printf("Enter a number (1 to 999): ");

scanf("%d", &number);

if (number < 1 || number > 999) {

printf("Number out of range. Please enter a number between 1 and 999.\n");

return 1;

}

printf("Number in words: ");

printNumberInWords(number);

return 0;

}

void printNumberInWords(int number) {

char \*ones[] = { "", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine" };

char \*teens[] = { "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen" };

char \*tens[] = { "", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety" };

if (number >= 100) {

printf("%s Hundred", ones[number / 100]);

number %= 100;

if (number > 0) {

printf(" and ");

}

}

if (number >= 20) {

printf("%s", tens[number / 10]);

number %= 10;

if (number > 0) {

printf("-%s", ones[number]);

}

} else if (number >= 10) {

printf("%s", teens[number - 10]);

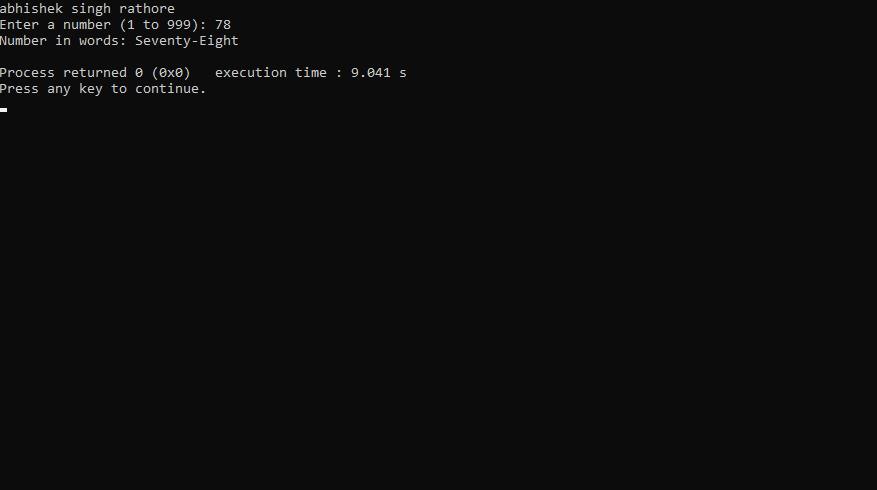
} else if (number > 0) {

printf("%s", ones[number]);

}

printf("\n");

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

printf("ASCII Characters and their Values:\n");

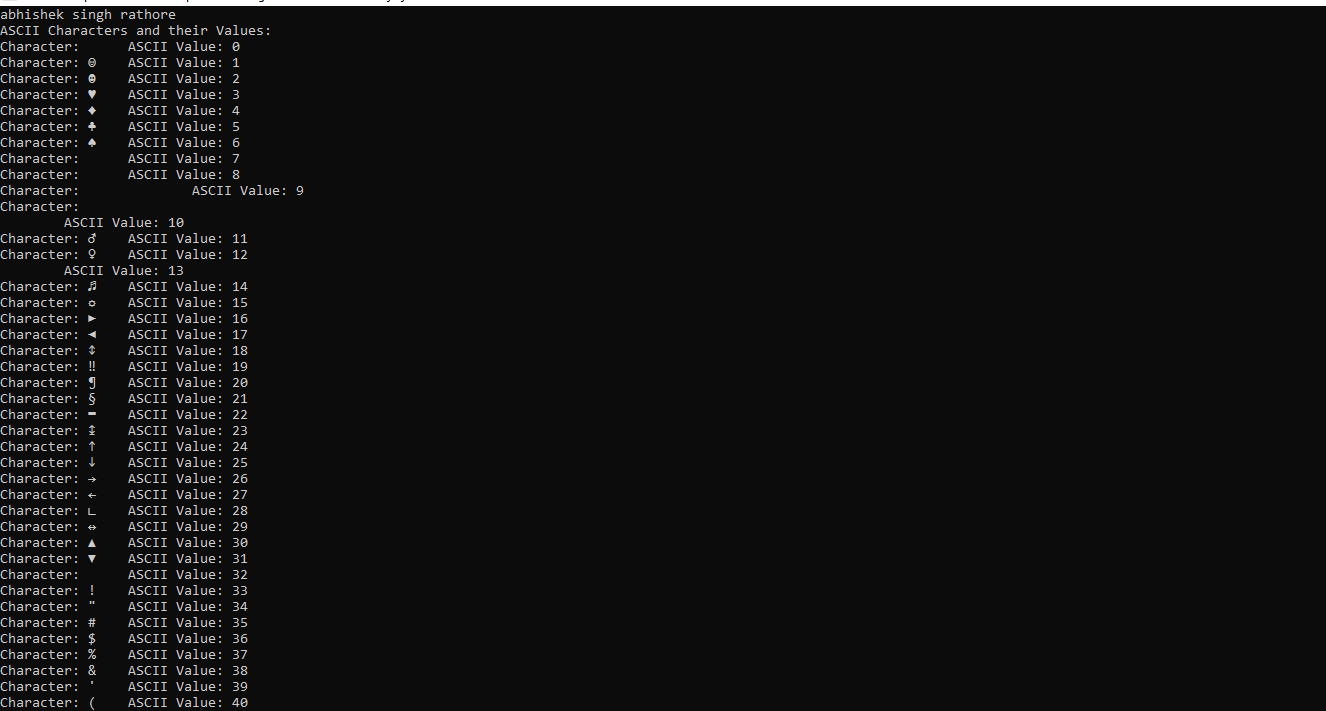
for (int i = 0; i <= 127; i++) {

printf("Character: %c\tASCII Value: %d\n", i, i);

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

double base, exponent, result = 1.0;

// Input the base and exponent

printf("Enter the base: ");

scanf("%lf", &base);

printf("Enter the exponent: ");

scanf("%lf", &exponent);

// Calculate the power using a for loop

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

result \*= base;

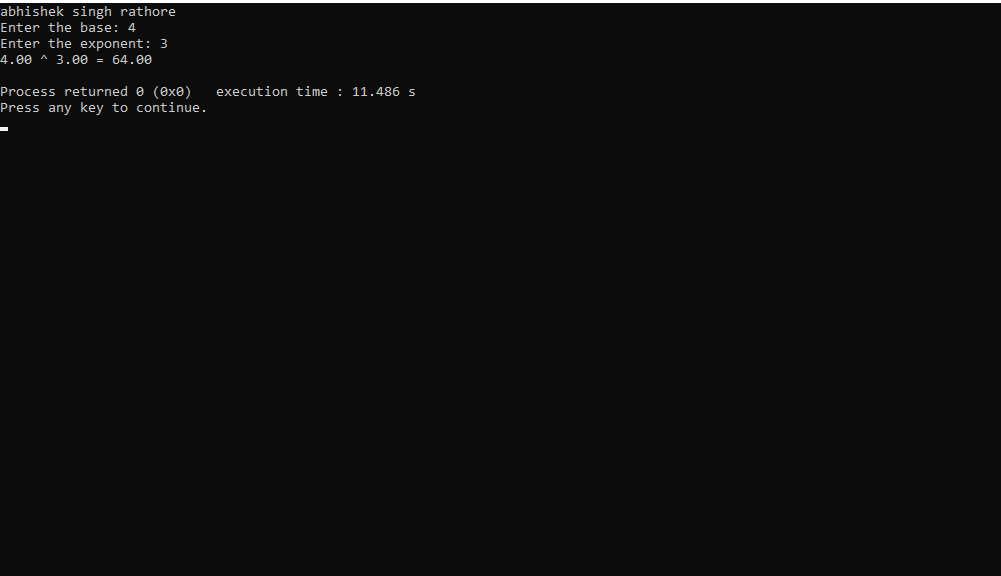
}

// Display the result

printf("%.2lf ^ %.2lf = %.2lf\n", base, exponent, result);

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore);");

int number;

// Input the number from the user

printf("Enter a positive integer: ");

scanf("%d", &number);

if (number <= 0) {

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

} else {

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

// Find and print the factors

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

if (number % i == 0) {

printf("%d", i);

if (i != number) {

printf(", ");

}

}

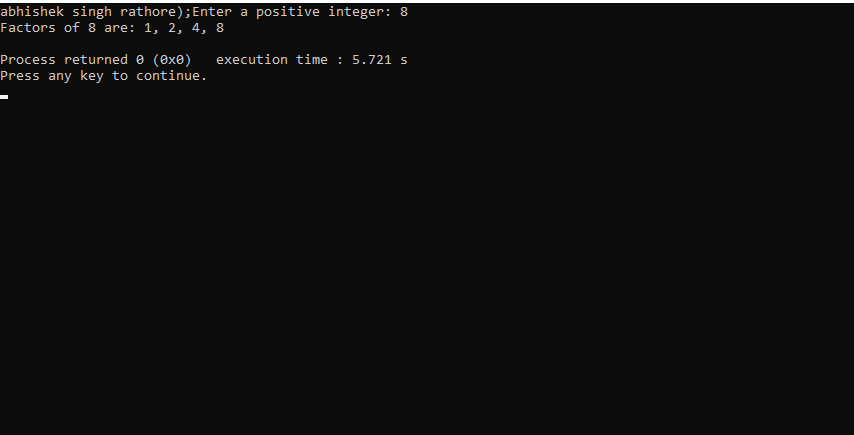
}

printf("\n");

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int num;

unsigned long long factorial = 1; // Use unsigned long long to handle larger factorials

printf("Enter a positive integer: ");

scanf("%d", &num);

// Check if the input is negative

if (num < 0) {

printf("Factorial is not defined for negative numbers.\n");

} else {

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

factorial \*= i;

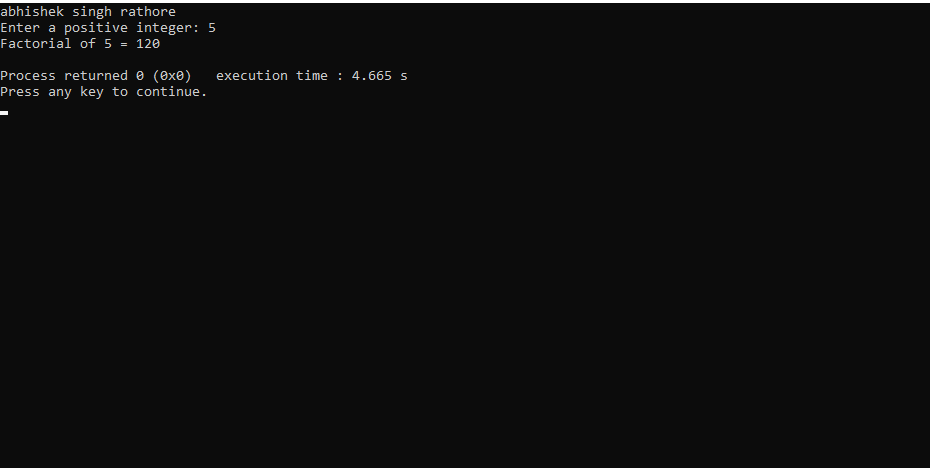
}

printf("Factorial of %d = %llu\n", num, factorial);

}

return 0;

}



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

#include <stdio.h>

// Function to find the HCF (GCD) of two numbers

int findHCF(int num1, int num2) {

while (num1 != num2) {

if (num1 > num2)

num1 -= num2;

else

num2 -= num1;

}

return num1;

}

int main() {

printf("abhishek singh rathore\n");

int num1, num2;

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

scanf("%d", &num2);

if (num1 < 0 || num2 < 0) {

printf("Please enter positive numbers.\n");

} else {

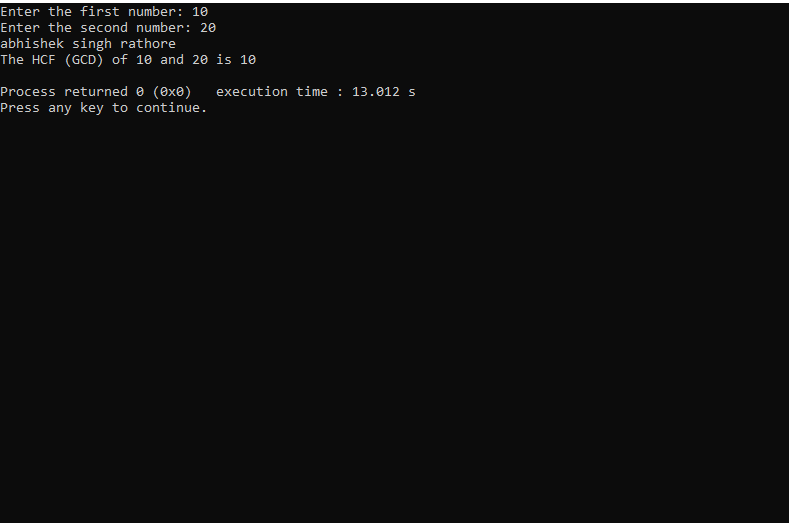
int hcf = findHCF(num1, num2);

printf("The HCF (GCD) of %d and %d is %d\n", num1, num2, hcf);

}

return 0;

}



**//47. Write a C program to find HCF (GCD) of two numbers.**

#include <stdio.h>

// Function to find the greatest common divisor (GCD) of two numbers

int findGCD(int a, int b) {

if (b == 0) {

return a;

}

return findGCD(b, a % b);

}

// Function to find the LCM of two numbers

int findLCM(int a, int b) {

int gcd = findGCD(a, b);

return (a \* b) / gcd;

}

int main() {

printf("abhishek singh rathore\n");

int num1, num2;

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

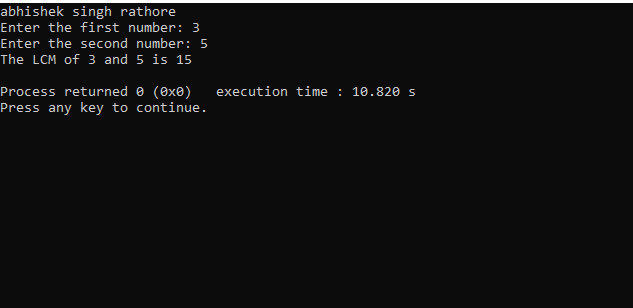
scanf("%d", &num2);

int lcm = findLCM(num1, num2);

printf("The LCM of %d and %d is %d\n", num1, num2, lcm);

return 0;

}



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

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int num, ele = 1; // Assume the number is prime initially

printf("Enter a number: ");

scanf("%d", &num);

if (num <= 1) {

ele = 0; // Numbers less than or equal to 1 are not prime

} else {

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0) {

ele = 0; // It's not prime if it's divisible by any number other than 1 and itself

break; // Exit the loop early since we already know it's not prime

}

}

}

if (ele) {

printf("%d is a prime number.\n");

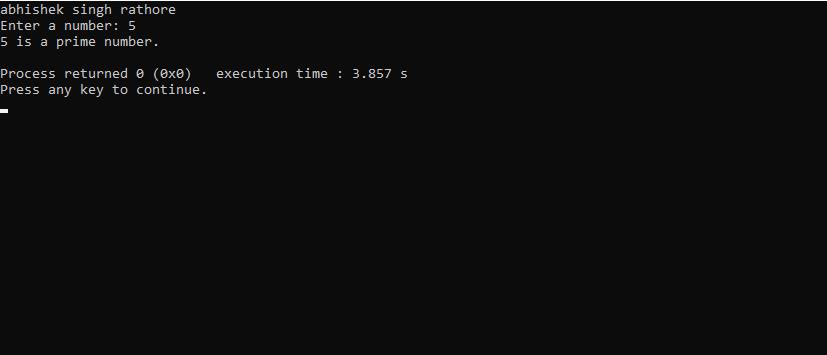
} else {

printf("%d is not a prime number.\n");

}

return 0;

}



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

#include <stdio.h>

int isPrime(int num) {

if (num <= 1) {

return 0;

}

if (num <= 3) {

return 1;

}

if (num % 2 == 0 || num % 3 == 0) {

return 0;

}

for (int i = 5; i \* i <= num; i += 6) {

if (num % i == 0 || num % (i + 2) == 0) {

return 0;

}

}

return 1;

}

int main() {

printf("abhishek singh rathore\n");

int n;

printf("Enter a positive integer (n): ");

scanf("%d", &n);

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

for (int i = 2; i <= n; i++) {

if (isPrime(i)) {

printf("%d ", i);

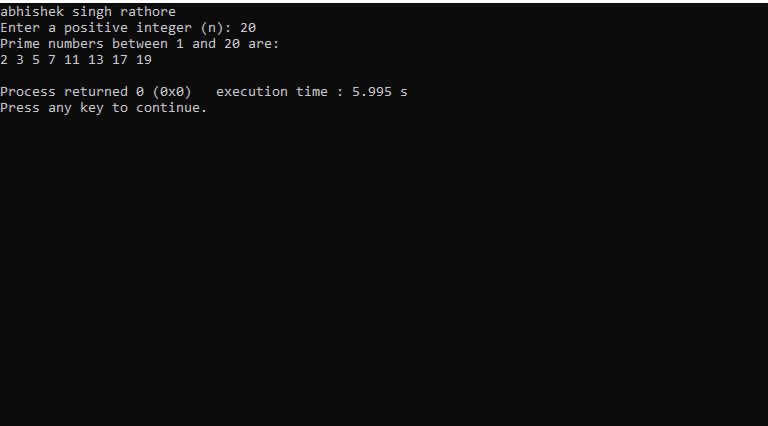
}

}

printf("\n");

return 0;

}



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

#include <stdio.h>

#include <stdbool.h>

bool isPrime(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

int main() {

int n, sum = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a positive integer (n): ");

scanf("%d", &n);

if (n <= 1) {

printf("There are no prime numbers in the specified range.\n");

} else {

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

for (int i = 2; i <= n; i++) {

if (isPrime(i)) {

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

sum += i;

}

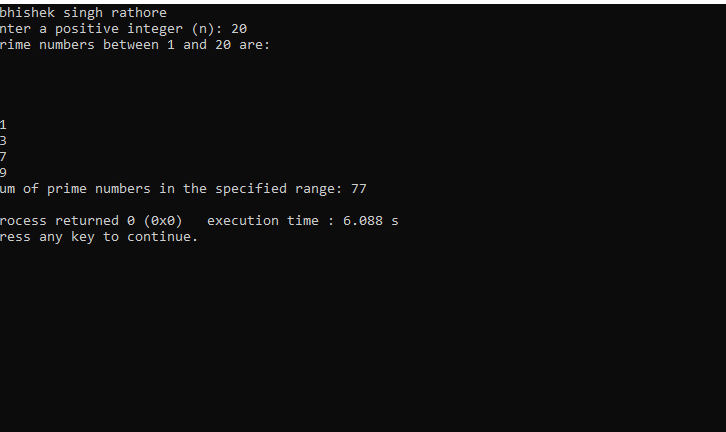
}

printf("Sum of prime numbers in the specified range: %d\n", sum);

}

return 0;

}



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

#include <stdio.h>

// Function to check if a number is prime

int isPrime(int num) {

if (num <= 1) {

return 0; // Not prime

}

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0) {

return 0; // Not prime

}

}

return 1; // Prime

}

int main() {

int number;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a positive integer: ");

scanf("%d", &number);

if (number <= 1) {

printf("Prime factors are not defined for numbers less than 2.\n");

} else {

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

// Find and print the prime factors

for (int i = 2; i <= number; i++) {

if (number % i == 0 && isPrime(i)) {

while (number % i == 0) {

printf("%d ", i);

number /= i;

}

}

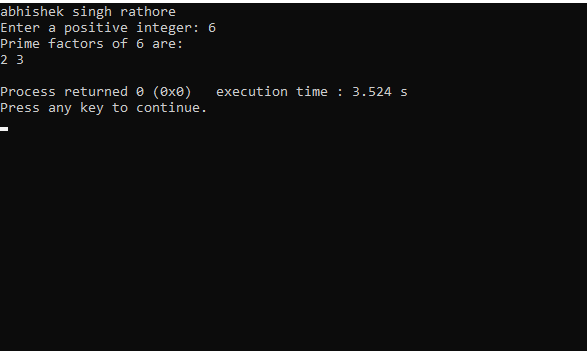
}

printf("\n");

}

return 0;

}



**// 52. 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, n = 0, result = 0;

originalNum = num;

// Count the number of digits

while (originalNum != 0) {

originalNum /= 10;

n++;

}

originalNum = num;

// Calculate the sum of digits each raised to the power of n

while (originalNum != 0) {

remainder = originalNum % 10;

result += pow(remainder, n);

originalNum /= 10;

}

return (result == num);

}

int main() {

int number;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a number: ");

scanf("%d", &number);

if (isArmstrong(number)) {

printf("%d is an Armstrong number.\n", number);

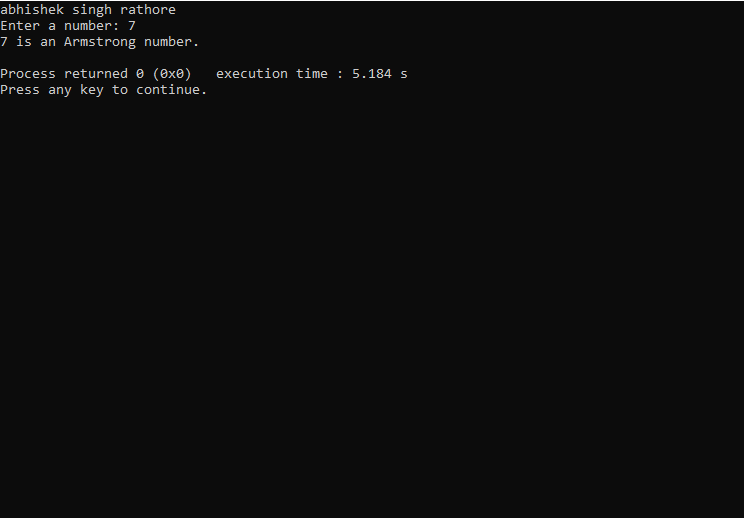
} else {

printf("%d is not an Armstrong number.\n", number);

}

return 0;

}



**//53. 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, n = 0, result = 0;

originalNum = num;

// Count the number of digits

while (originalNum != 0) {

originalNum /= 10;

n++;

}

originalNum = num;

// Calculate the sum of digits each raised to the power of n

while (originalNum != 0) {

remainder = originalNum % 10;

result += pow(remainder, n);

originalNum /= 10;

}

return (result == num);

}

int main() {

int n;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a positive integer (n): ");

scanf("%d", &n);

if (n < 1) {

printf("Armstrong numbers are not defined for numbers less than 1.\n");

} else {

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

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

if (isArmstrong(i)) {

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

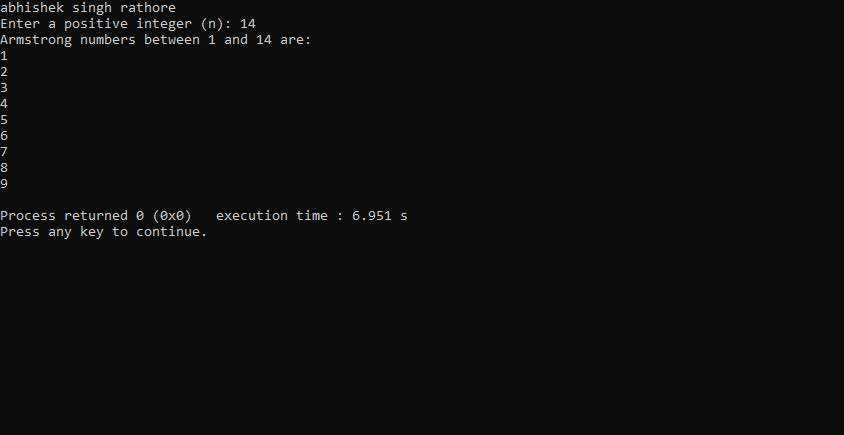
}

}

}

return 0;

}



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

#include <stdio.h>

int isPerfect(int num) {

int sum = 0;

// Find the divisors and calculate the sum of proper divisors

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

if (num % i == 0) {

sum += i;

}

}

return (sum == num);

}

int main() {

int number;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a positive integer: ");

scanf("%d", &number);

if (number <= 0) {

printf("Perfect numbers are defined for positive integers only.\n");

} else if (isPerfect(number)) {

printf("%d is a perfect number.\n", number);

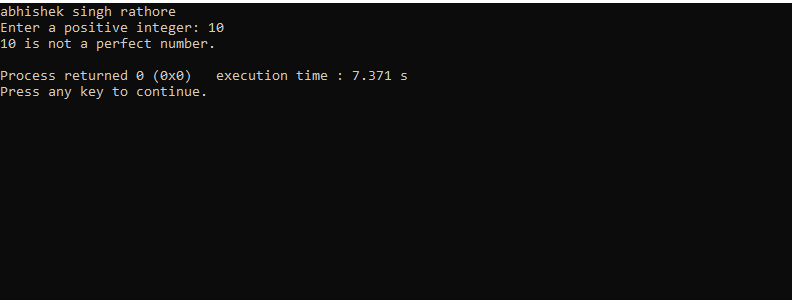
} else {

printf("%d is not a perfect number.\n", number);

}

return 0;

}



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

#include <stdio.h>

int isPerfect(int num) {

int sum = 0;

// Find the divisors and calculate the sum of proper divisors

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

if (num % i == 0) {

sum += i;

}

}

return (sum == num);

}

int main() {

int n;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a positive integer (n): ");

scanf("%d", &n);

if (n <= 0) {

printf("Perfect numbers are defined for positive integers only.\n");

} else {

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

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

if (isPerfect(i)) {

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

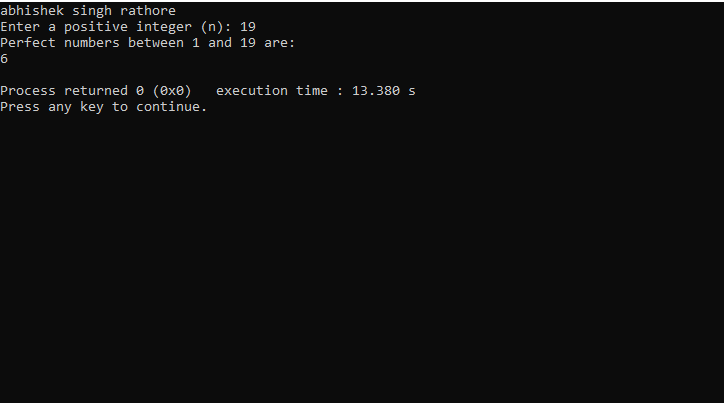
}

}

}

return 0;

}



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

#include <stdio.h>

int factorial(int num) {

if (num == 0) {

return 1;

} else {

int fact = 1;

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

fact \*= i;

}

return fact;

}

}

int isStrong(int num) {

int originalNum, remainder, sum = 0;

originalNum = num;

// Calculate the sum of the factorial of digits

while (originalNum != 0) {

remainder = originalNum % 10;

sum += factorial(remainder);

originalNum /= 10;

}

return (sum == num);

}

int main() {

int number;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a positive integer: ");

scanf("%d", &number);

if (number <= 0) {

printf("Strong numbers are defined for positive integers only.\n");

} else if (isStrong(number)) {

printf("%d is a strong number.\n", number);

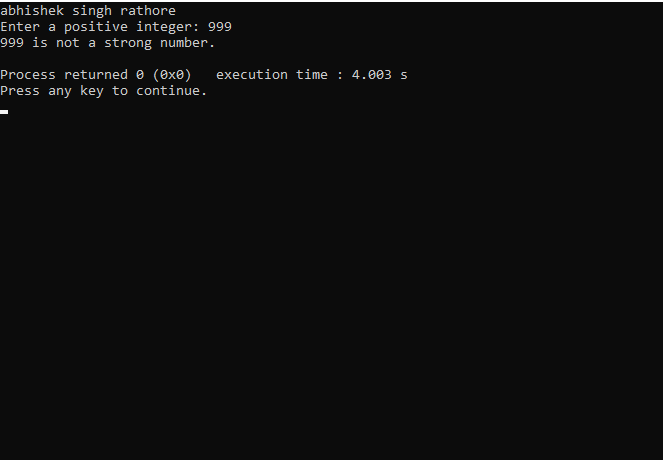
} else {

printf("%d is not a strong number.\n", number);

}

return 0;

}



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

#include <stdio.h>

int factorial(int num)

{

if (num == 0) {

return 1;

} else {

int fact = 1;

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

fact \*= i;

}

return fact;

}

}

int isStrong(int num) {

int originalNum, remainder, sum = 0;

originalNum = num;

// Calculate the sum of the factorial of digits

while (originalNum != 0) {

remainder = originalNum % 10;

sum += factorial(remainder);

originalNum /= 10;

}

return (sum == num);

}

int main() {

int n;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a positive integer (n): ");

scanf("%d", &n);

if (n <= 0) {

printf("Strong numbers are defined for positive integers only.\n");

} else {

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

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

if (isStrong(i)) {

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

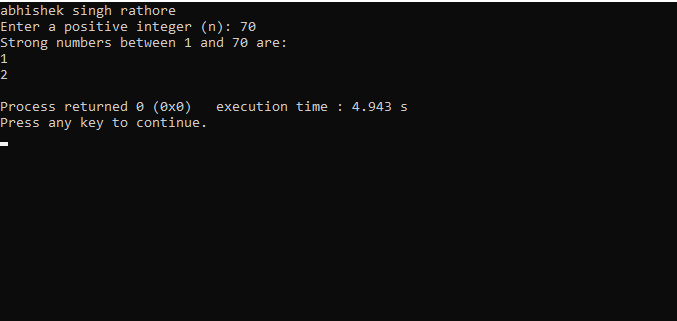
}

}

}

return 0;

}



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

#include <stdio.h>

int main() {

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

// Ask the user for input

printf("abhishek singh rathore\n");

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

scanf("%d", &n);

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

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

if (i <= 1) {

next = i;

} else {

next = first + second;

first = second;

second = next;

}

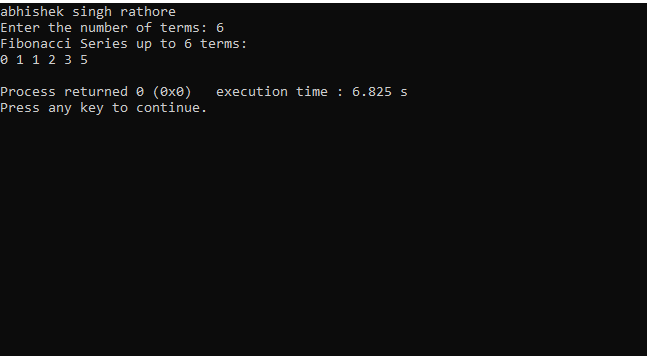
printf("%d ", next);

}

printf("\n");

return 0;

}



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

#include <stdio.h>

int main() {

char binary[32];

char onesComplement[32];

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a binary number: ");

scanf("%s", binary);

// Calculate the one's complement

int i = 0;

while (binary[i] != '\0') {

if (binary[i] == '0') {

onesComplement[i] = '1';

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

onesComplement[i] = '0';

} else {

printf("Invalid binary number.\n");

return 1;

}

i++;

}

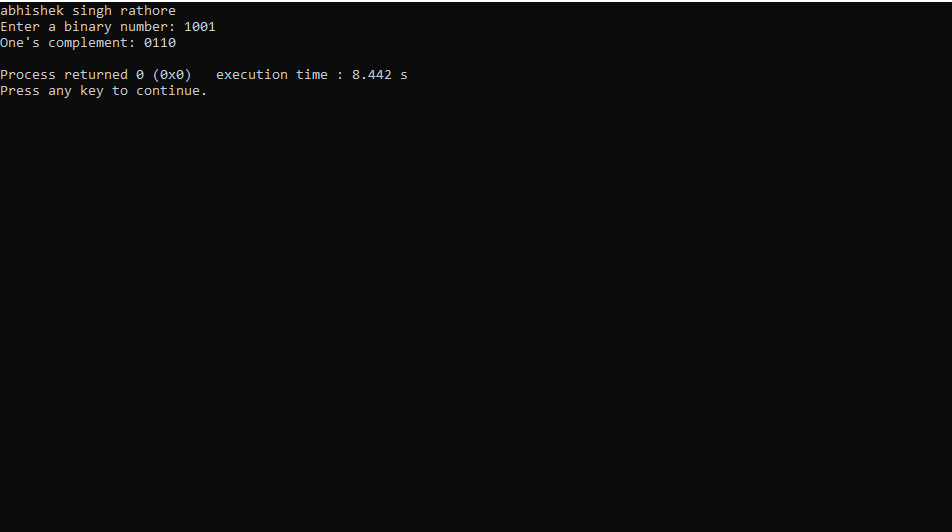
onesComplement[i] = '\0';

// Print the one's complement

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

return 0;

}



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

#include <stdio.h>

void reverse(char binary[]) {

int start = 0;

int end = strlen(binary) - 1;

while (start < end) {

char temp = binary[start];

binary[start] = binary[end];

binary[end] = temp;

start++;

end--;

}

}

void onesComplement(char binary[]) {

for (int i = 0; binary[i] != '\0'; i++) {

if (binary[i] == '0') {

binary[i] = '1';

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

binary[i] = '0';

}

}

}

int main() {

char binary[32];

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a binary number: ");

scanf("%s", binary);

// Reverse the binary number

reverse(binary);

// Calculate the one's complement

onesComplement(binary);

// Add 1 to the LSB to find the two's complement

int carry = 1;

for (int i = 0; binary[i] != '\0'; i++) {

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

binary[i] = '1';

carry = 0;

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

binary[i] = '0';

}

}

// Reverse the result to obtain two's complement

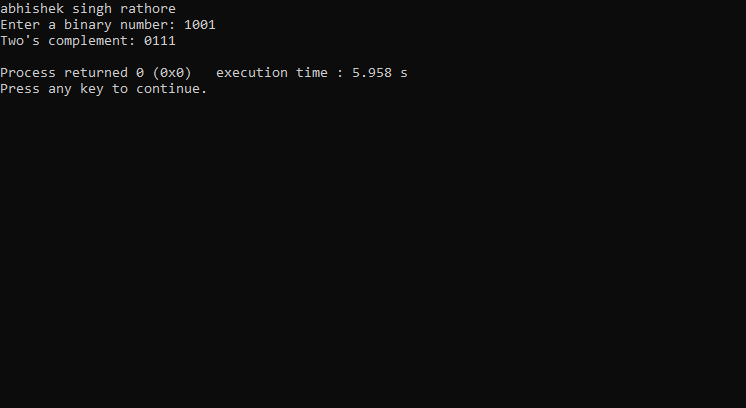
reverse(binary);

// Print the two's complement

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

return 0;

}



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

#include <stdio.h>

#include <math.h>

// Function to convert a binary digit to an octal digit

int binaryToOctalDigit(char binaryDigit) {

int decimalDigit = binaryDigit - '0'; // Convert ASCII '0' or '1' to integer 0 or 1

return decimalDigit;

}

int main() {

char binary[32];

int octal[32];

int length, octalDigit, i, j;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a binary number: ");

scanf("%s", binary);

// Calculate the length of the binary number

length = strlen(binary);

// Make sure the binary length is a multiple of 3 by adding leading zeros if needed

int remainder = length % 3;

if (remainder > 0) {

for (i = 0; i < 3 - remainder; i++) {

binary[length + i] = '0';

}

length += 3 - remainder;

}

// Initialize the octal array to zero

for (i = 0; i < length / 3; i++) {

octal[i] = 0;

}

// Convert binary to octal

for (i = 0, j = length / 3 - 1; i < length; i += 3, j--) {

octalDigit = binaryToOctalDigit(binary[i]) \* 4 +

binaryToOctalDigit(binary[i + 1]) \* 2 +

binaryToOctalDigit(binary[i + 2]);

octal[j] = octalDigit;

}

// Print the octal number

printf("Octal equivalent: ");

for (i = 0; i < length / 3; i++) {

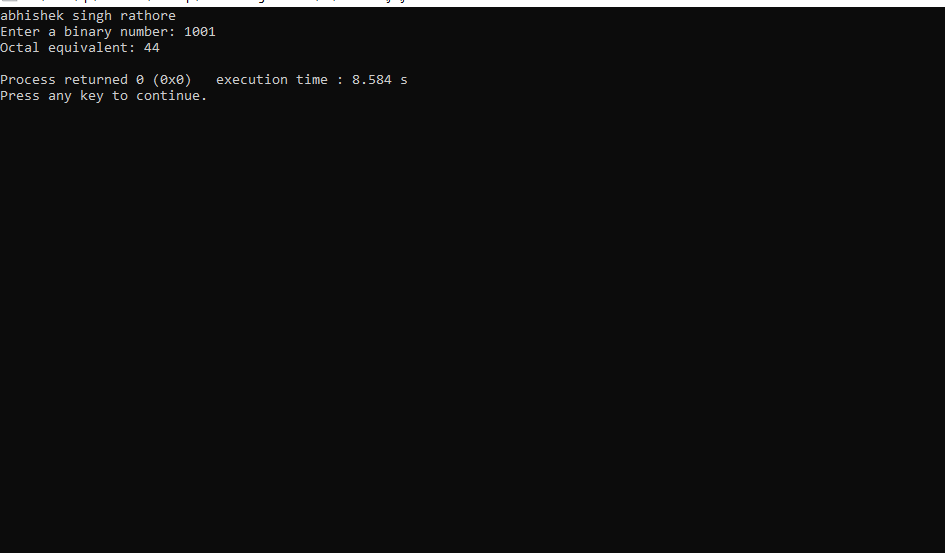
printf("%d", octal[i]);

}

printf("\n");

return 0;

}



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

#include <stdio.h>

#include <string.h>

int main() {

char binary[32];

int decimal = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a binary number: ");

scanf("%s", binary);

int length = strlen(binary);

int base = 1; // 2^0

// Convert binary to decimal

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

if (binary[i] == '1') {

decimal += base;

}

base \*= 2; // Multiply by 2 for the next position

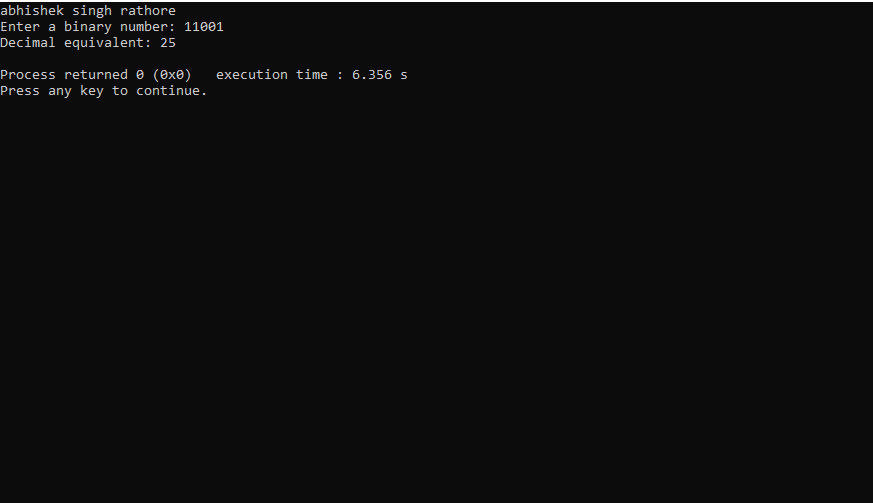
}

// Print the decimal number

printf("Decimal equivalent: %d\n", decimal);

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert a 4-bit binary number to a hexadecimal digit

char binaryToHexDigit(char binaryDigits[4]) {

int decimal = 0;

for (int i = 0; i < 4; i++) {

decimal = (decimal << 1) | (binaryDigits[i] - '0');

}

if (decimal >= 0 && decimal <= 9) {

return (char)(decimal + '0');

} else {

return (char)(decimal - 10 + 'A');

}

}

int main() {

char binary[32];

char hex[32];

int length, hexDigitCount = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a binary number: ");

scanf("%s", binary);

// Calculate the length of the binary number

length = strlen(binary);

// Make sure the binary length is a multiple of 4 by adding leading zeros if needed

int remainder = length % 4;

if (remainder > 0) {

for (int i = 0; i < 4 - remainder; i++) {

binary[length + i] = '0';

}

length += 4 - remainder;

}

// Convert binary to hexadecimal

for (int i = 0; i < length; i += 4) {

char binaryDigits[4];

for (int j = 0; j < 4; j++) {

binaryDigits[j] = binary[i + j];

}

hex[hexDigitCount] = binaryToHexDigit(binaryDigits);

hexDigitCount++;

}

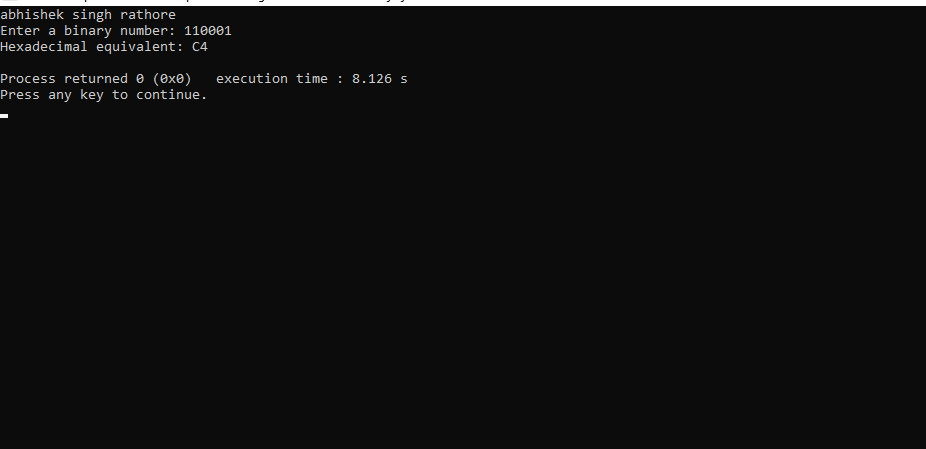
hex[hexDigitCount] = '\0';

// Print the hexadecimal number

printf("Hexadecimal equivalent: %s\n", hex);

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert an octal digit to its binary representation

void octalToBinaryDigit(char octalDigit, char binaryDigits[4]) {

int decimal = octalDigit - '0';

int index = 0;

// Convert octal to binary

while (decimal > 0) {

binaryDigits[index] = (decimal % 2) + '0';

decimal /= 2;

index++;

}

// Pad with leading zeros to make it 3 bits

while (index < 3) {

binaryDigits[index] = '0';

index++;

}

// Null-terminate the string

binaryDigits[3] = '\0';

// Reverse the binaryDigits array to get the correct order

char temp;

for (int i = 0; i < index / 2; i++) {

temp = binaryDigits[i];

binaryDigits[i] = binaryDigits[index - i - 1];

binaryDigits[index - i - 1] = temp;

}

}

int main() {

char octal[32];

char binary[96]; // Maximum 32 octal digits can be converted to 96 binary digits

int length, binaryDigitCount = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter an octal number: ");

scanf("%s", octal);

// Calculate the length of the octal number

length = strlen(octal);

// Convert octal to binary

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

char binaryDigits[4];

octalToBinaryDigit(octal[i], binaryDigits);

strcat(binary, binaryDigits);

binaryDigitCount += 3;

}

// Remove leading zeros if any

int leadingZeroCount = 0;

while (binary[leadingZeroCount] == '0') {

leadingZeroCount++;

}

// Print the binary number

if (binary[leadingZeroCount] == '\0') {

printf("Binary equivalent: 0\n");

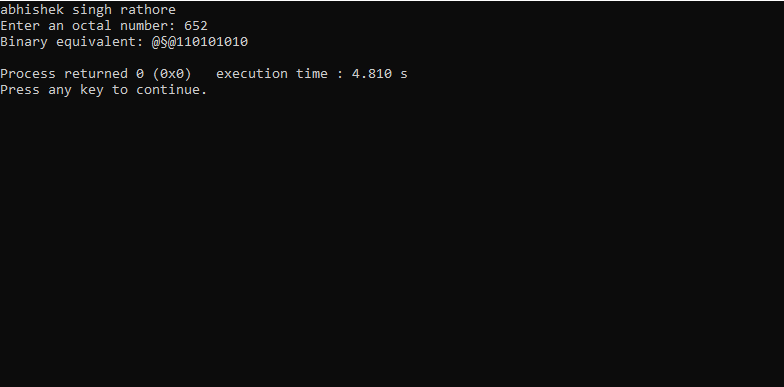
} else {

printf("Binary equivalent: %s\n", binary + leadingZeroCount);

}

return 0;

}



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

#include <stdio.h>

#include <string.h>

int main() {

char octal[32];

int decimal = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter an octal number: ");

scanf("%s", octal);

int length = strlen(octal);

int base = 1; // 8^0

// Convert octal to decimal

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

int octalDigit = octal[i] - '0';

decimal += octalDigit \* base;

base \*= 8; // Multiply by 8 for the next position

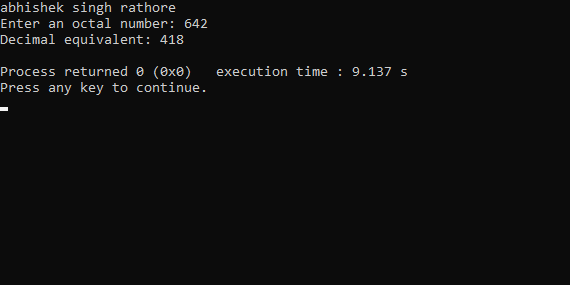
}

// Print the decimal number

printf("Decimal equivalent: %d\n", decimal);

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert an octal digit to its binary representation

void octalToBinaryDigit(char octalDigit, char binaryDigits[4]) {

int decimal = octalDigit - '0';

int index = 0;

// Convert octal to binary

while (decimal > 0) {

binaryDigits[index] = (decimal % 2) + '0';

decimal /= 2;

index++;

}

// Pad with leading zeros to make it 3 bits

while (index < 3) {

binaryDigits[index] = '0';

index++;

}

// Null-terminate the string

binaryDigits[3] = '\0';

// Reverse the binaryDigits array to get the correct order

char temp;

for (int i = 0; i < index / 2; i++) {

temp = binaryDigits[i];

binaryDigits[i] = binaryDigits[index - i - 1];

binaryDigits[index - i - 1] = temp;

}

}

// Function to convert a 4-bit binary number to a hexadecimal digit

char binaryToHexDigit(char binaryDigits[4]) {

int decimal = 0;

for (int i = 0; i < 4; i++) {

decimal = (decimal << 1) | (binaryDigits[i] - '0');

}

if (decimal >= 0 && decimal <= 9) {

return (char)(decimal + '0');

} else {

return (char)(decimal - 10 + 'A');

}

}

int main() {

char octal[32];

char binary[96]; // Maximum 32 octal digits can be converted to 96 binary digits

char hex[32]; // Maximum 32 octal digits can be converted to 8 hexadecimal digits

int length, binaryDigitCount = 0, hexDigitCount = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter an octal number: ");

scanf("%s", octal);

// Calculate the length of the octal number

length = strlen(octal);

// Convert octal to binary

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

char binaryDigits[4];

octalToBinaryDigit(octal[i], binaryDigits);

strcat(binary, binaryDigits);

binaryDigitCount += 3;

}

// Make sure the binary length is a multiple of 4 by adding leading zeros if needed

int remainder = binaryDigitCount % 4;

if (remainder > 0) {

for (int i = 0; i < 4 - remainder; i++) {

binary[binaryDigitCount + i] = '0';

}

binaryDigitCount += 4 - remainder;

}

// Convert binary to hexadecimal

for (int i = 0; i < binaryDigitCount; i += 4) {

char binaryDigits[4];

for (int j = 0; j < 4; j++) {

binaryDigits[j] = binary[i + j];

}

hex[hexDigitCount] = binaryToHexDigit(binaryDigits);

hexDigitCount++;

}

hex[hexDigitCount] = '\0';

// Remove leading zeros if any

int leadingZeroCount = 0;

while (hex[leadingZeroCount] == '0') {

leadingZeroCount++;

}

// Print the hexadecimal number

if (hex[leadingZeroCount] == '\0') {

printf("Hexadecimal equivalent: 0\n");

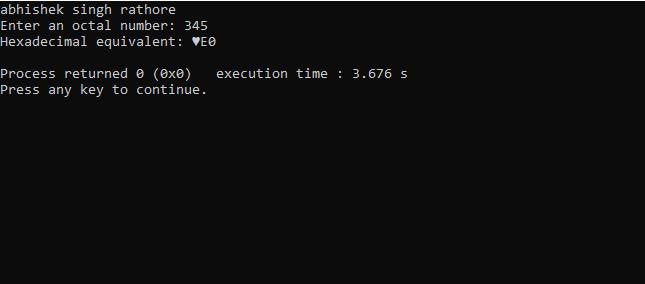
} else {

printf("Hexadecimal equivalent: %s\n", hex + leadingZeroCount);

}

return 0;

}



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

#include <stdio.h>

int main() {

int decimal;

int binary[32]; // To store the binary digits

int i = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a decimal number: ");

scanf("%d", &decimal);

if (decimal < 0) {

printf("Binary representation is not defined for negative numbers.\n");

return 1;

}

// Convert decimal to binary

if (decimal == 0) {

binary[i] = 0;

i++;

} else {

while (decimal > 0) {

binary[i] = decimal % 2;

decimal = decimal / 2;

i++;

}

}

// Print the binary representation

printf("Binary equivalent: ");

if (i == 0) {

printf("0"); // Handle the case when the input is 0

} else {

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

printf("%d", binary[j]);

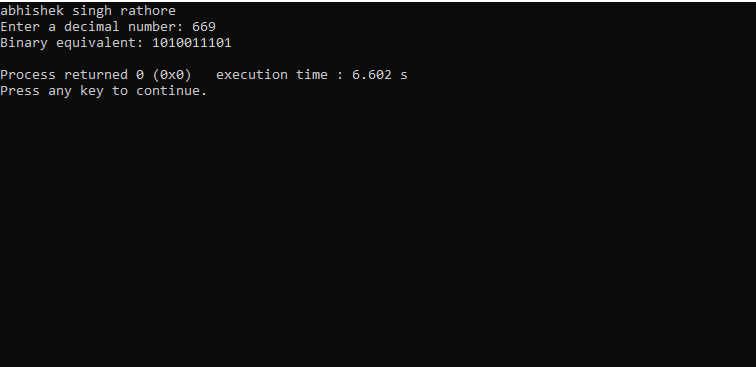
}

}

printf("\n");

return 0;

}



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

#include <stdio.h>

int main() {

int decimal;

int octal[32]; // To store the octal digits

int i = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a decimal number: ");

scanf("%d", &decimal);

if (decimal < 0) {

printf("Octal representation is not defined for negative numbers.\n");

return 1;

}

// Convert decimal to octal

if (decimal == 0) {

octal[i] = 0;

i++;

} else {

while (decimal > 0) {

octal[i] = decimal % 8;

decimal = decimal / 8;

i++;

}

}

// Print the octal representation

printf("Octal equivalent: ");

if (i == 0) {

printf("0"); // Handle the case when the input is 0

} else {

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

printf("%d", octal[j]);

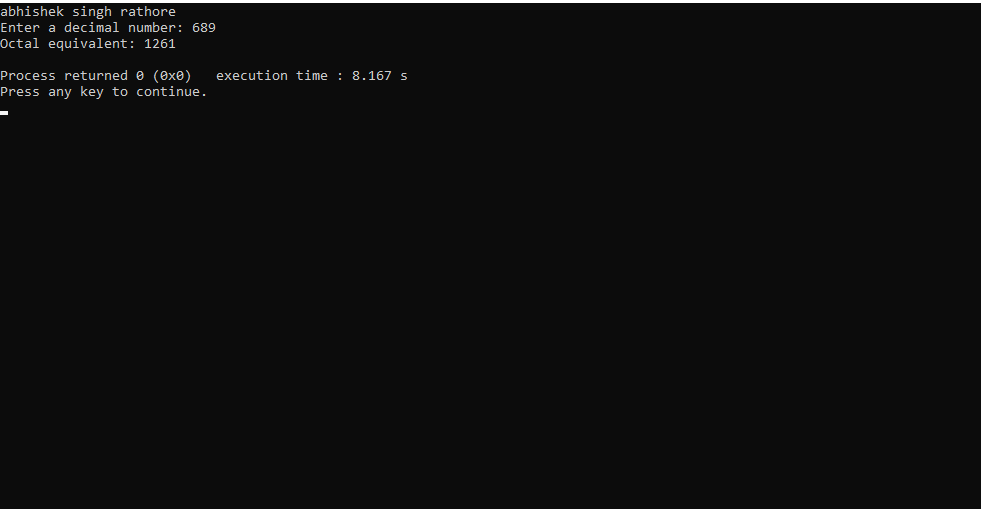
}

}

printf("\n");

return 0;

}



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

#include <stdio.h>

int main() {

int decimal;

char hexadecimal[32]; // To store the hexadecimal digits

int i = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a decimal number: ");

scanf("%d", &decimal);

if (decimal < 0) {

printf("Hexadecimal representation is not defined for negative numbers.\n");

return 1;

}

// Convert decimal to hexadecimal

if (decimal == 0) {

hexadecimal[i] = '0';

i++;

} else {

while (decimal > 0) {

int remainder = decimal % 16;

if (remainder < 10) {

hexadecimal[i] = remainder + '0';

} else {

hexadecimal[i] = remainder - 10 + 'A';

}

decimal = decimal / 16;

i++;

}

}

// Print the hexadecimal representation

printf("Hexadecimal equivalent: ");

if (i == 0) {

printf("0"); // Handle the case when the input is 0

} else {

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

printf("%c", hexadecimal[j]);

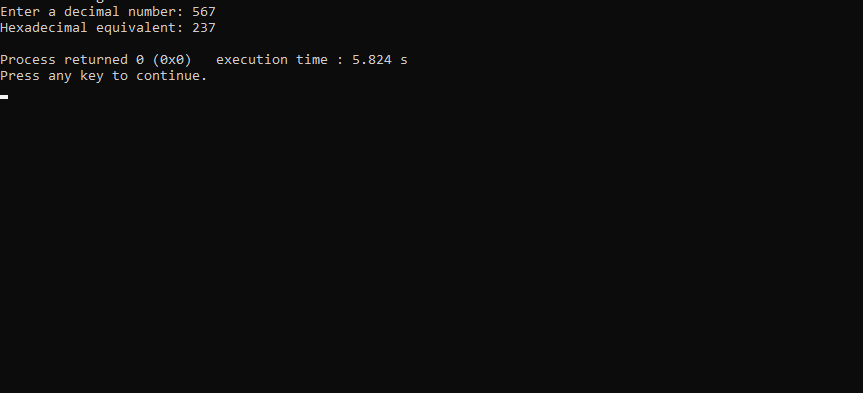
}

}

printf("\n");

return 0;

}



**//Write a C program to convert Hexadecimal to Binary number system.**

#include <stdio.h>

#include <string.h>

// Function to convert a hexadecimal digit to its binary representation

void hexToBinaryDigit(char hexDigit, char binaryDigits[5]) {

int decimal;

if (hexDigit >= '0' && hexDigit <= '9') {

decimal = hexDigit - '0';

} else if (hexDigit >= 'A' && hexDigit <= 'F') {

decimal = hexDigit - 'A' + 10;

} else if (hexDigit >= 'a' && hexDigit <= 'f') {

decimal = hexDigit - 'a' + 10;

} else {

printf("Invalid hexadecimal digit: %c\n", hexDigit);

return;

}

// Convert decimal to binary

int index = 0;

while (decimal > 0) {

binaryDigits[index] = (decimal % 2) + '0';

decimal /= 2;

index++;

}

// Pad with leading zeros to make it 4 bits

while (index < 4) {

binaryDigits[index] = '0';

index++;

}

// Null-terminate the string

binaryDigits[4] = '\0';

// Reverse the binaryDigits array to get the correct order

char temp;

for (int i = 0; i < index / 2; i++) {

temp = binaryDigits[i];

binaryDigits[i] = binaryDigits[index - i - 1];

binaryDigits[index - i - 1] = temp;

}

}

int main() {

char hexadecimal[32];

char binary[128]; // Maximum 32 hexadecimal digits can be converted to 128 binary digits

int length, binaryDigitCount = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a hexadecimal number: ");

scanf("%s", hexadecimal);

// Calculate the length of the hexadecimal number

length = strlen(hexadecimal);

// Convert hexadecimal to binary

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

char binaryDigits[5];

hexToBinaryDigit(hexadecimal[i], binaryDigits);

strcat(binary, binaryDigits);

binaryDigitCount += 4;

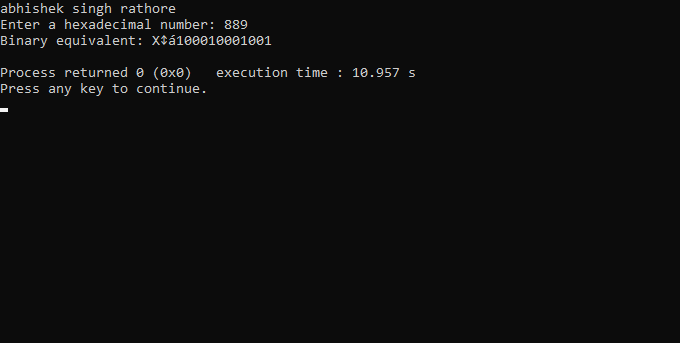
}

// Print the binary representation

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

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert a hexadecimal digit to its 4-bit binary representation

void hexToBinaryDigit(char hexDigit, char binaryDigits[5]) {

int decimal;

if (hexDigit >= '0' && hexDigit <= '9') {

decimal = hexDigit - '0';

} else if (hexDigit >= 'A' && hexDigit <= 'F') {

decimal = hexDigit - 'A' + 10;

} else if (hexDigit >= 'a' && hexDigit <= 'f') {

decimal = hexDigit - 'a' + 10;

} else {

printf("Invalid hexadecimal digit: %c\n", hexDigit);

return;

}

// Convert decimal to binary

int index = 0;

while (decimal > 0) {

binaryDigits[index] = (decimal % 2) + '0';

decimal /= 2;

index++;

}

// Pad with leading zeros to make it 4 bits

while (index < 4) {

binaryDigits[index] = '0';

index++;

}

// Null-terminate the string

binaryDigits[4] = '\0';

// Reverse the binaryDigits array to get the correct order

char temp;

for (int i = 0; i < index / 2; i++) {

temp = binaryDigits[i];

binaryDigits[i] = binaryDigits[index - i - 1];

binaryDigits[index - i - 1] = temp;

}

}

int main() {

char hexadecimal[32];

char binary[128]; // Maximum 32 hexadecimal digits can be converted to 128 binary digits

char octal[128]; // Maximum 32 hexadecimal digits can be converted to 42 octal digits

int length, binaryDigitCount = 0, octalDigitCount = 0;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a hexadecimal number: ");

scanf("%s", hexadecimal);

// Calculate the length of the hexadecimal number

length = strlen(hexadecimal);

// Convert hexadecimal to binary

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

char binaryDigits[5];

hexToBinaryDigit(hexadecimal[i], binaryDigits);

strcat(binary, binaryDigits);

binaryDigitCount += 4;

}

// Make sure the binary length is a multiple of 3 by adding leading zeros if needed

int remainder = binaryDigitCount % 3;

if (remainder > 0) {

for (int i = 0; i < 3 - remainder; i++) {

binary[binaryDigitCount + i] = '0';

}

binaryDigitCount += 3 - remainder;

}

// Convert binary to octal

for (int i = 0; i < binaryDigitCount; i += 3) {

char binaryDigits[4];

for (int j = 0; j < 3; j++) {

binaryDigits[j] = binary[i + j];

}

binaryDigits[3] = '\0';

int decimal = 0;

for (int j = 0; j < 3; j++) {

decimal = (decimal << 1) | (binaryDigits[j] - '0');

}

octal[octalDigitCount] = decimal + '0';

octalDigitCount++;

}

octal[octalDigitCount] = '\0';

// Remove leading zeros if any

int leadingZeroCount = 0;

while (octal[leadingZeroCount] == '0') {

leadingZeroCount++;

}

// Print the octal representation

if (octal[leadingZeroCount] == '\0') {

printf("Octal equivalent: 0\n");

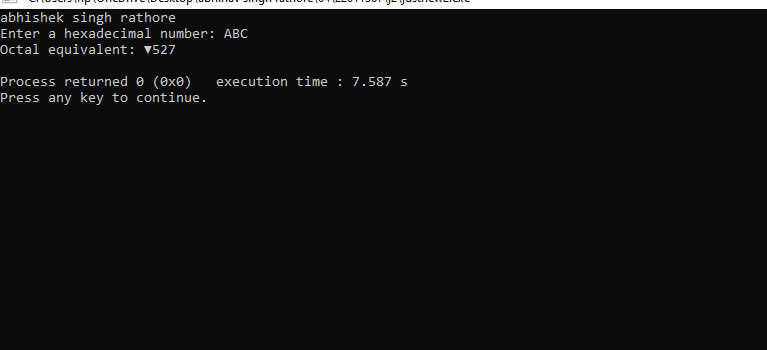
} else {

printf("Octal equivalent: %s\n", octal + leadingZeroCount);

}

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert a hexadecimal digit to its decimal value

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 {

printf("Invalid hexadecimal digit: %c\n", hexDigit);

return -1; // Return an error value

}

}

int main() {

char hexadecimal[32];

int decimal = 0;

int length;

// Ask the user for input

printf("abhishek singh rathore\n");

printf("Enter a hexadecimal number: ");

scanf("%s", hexadecimal);

// Calculate the length of the hexadecimal number

length = strlen(hexadecimal);

// Convert hexadecimal to decimal

for (int i = length - 1, power = 0; i >= 0; i--, power++) {

int digitValue = hexToDecimalDigit(hexadecimal[i]);

if (digitValue == -1) {

// Handle invalid digit

return 1;

}

decimal += digitValue \* (1 << (4 \* power)); // Equivalent to 16^power

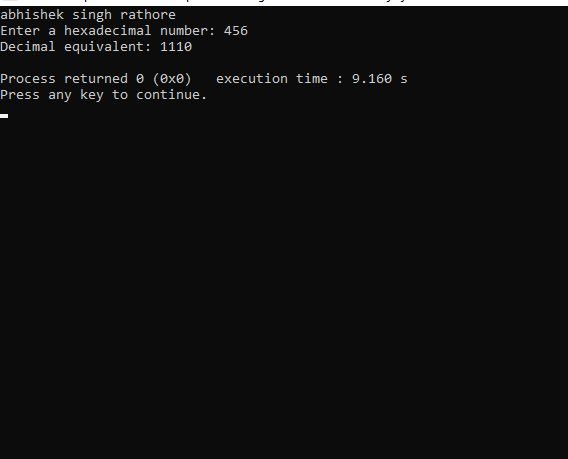
}

// Print the decimal number

printf("Decimal equivalent: %d\n", decimal);

return 0;

}



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

**1. Pyramid star pattern**

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int rows, space, star;

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

scanf("%d", &rows);

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

// Print spaces

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

printf(" ");

}

// Print stars

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

printf("\*");

}

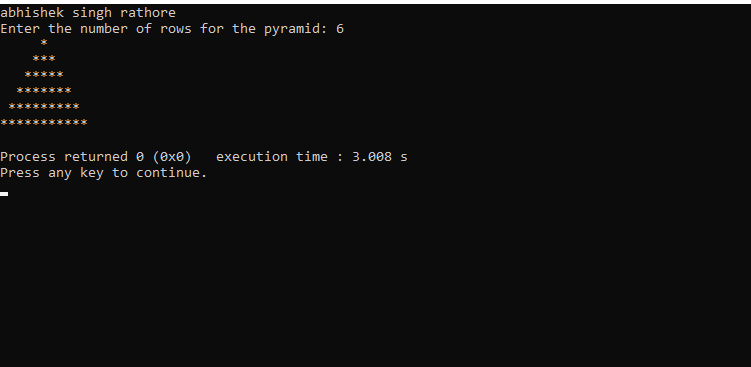
// Move to the next line

printf("\n");

}

return 0;

}



**2.Hollow pyramid**

#include <stdio.h>

int main() {

printf("abhishek singh rathore\n");

int i, j, rows;

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

scanf("%d", &rows);

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

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

printf(" ");

}

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

if (j == 1 || j == (2 \* i - 1) || i == rows) {

printf("\*");

} else {

printf(" ");

}

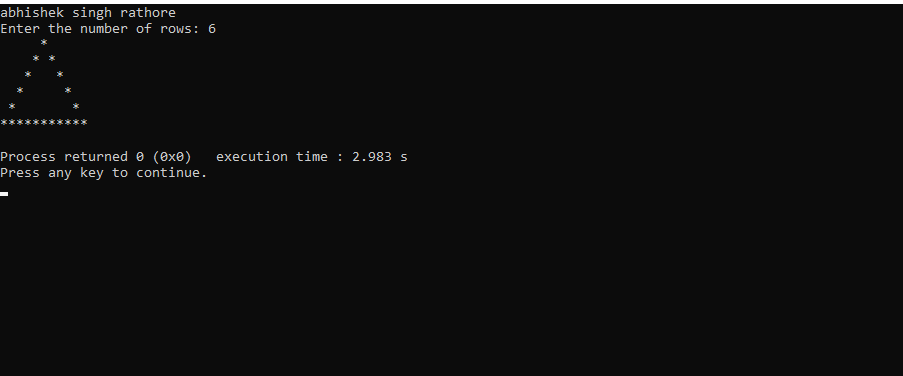
}

printf("\n");

}

return 0;

}



**3. Inverted pyramid star pattern.**

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int i, j, rows;

printf(" ");

printf("\nEnter number of rows : ");

scanf("%d", &rows);

for(i=1; i<=rows; i++)

{

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

{

printf(" ");

}

for(j=1; j<=(rows\*2 -(2\*i-1)); j++)

{

printf("\*");

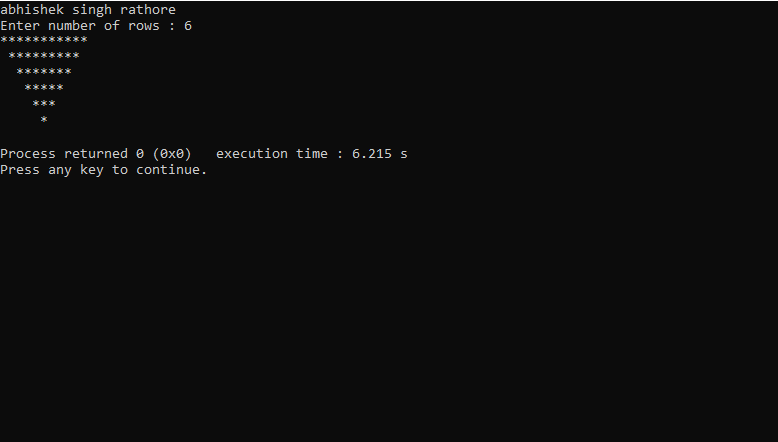
}

printf("\n");

}

return 0;

}



**4.Hollow Inverted star pattern.**

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int i, j, rows;

printf(" ");

printf("\nEnter number of rows: ");

scanf("%d", &rows);

for(i=1; i<=rows; i++)

{

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

{

printf(" ");

}

for(j=1; j<=(rows\*2 - (2\*i-1)); j++)

{

if(i==1 || j==1 || j==(rows\*2 - (2\*i - 1)))

{

printf("\*");

}

else

{

printf(" ");

}

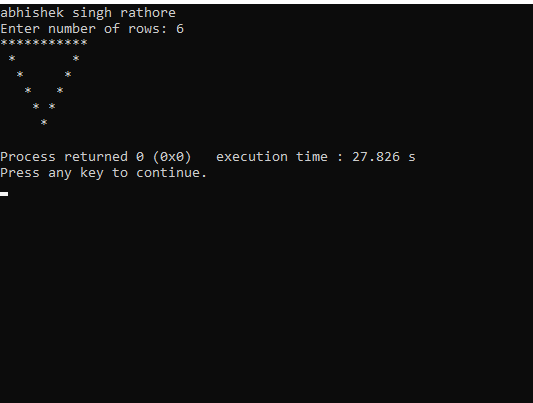
}

printf("\n");

}

return 0;

}



**5.Half diamond star pattern**

#include<stdio.h>

int main()

{

printf("abhishek singh rathore");

int i, j, N, columns;

printf(" ");

printf("\nEnter number of columns:");

scanf("%d",&N);

columns=1;

for(i=1;i<N\*2;i++)

{

for(j=1; j<=columns; j++)

{

printf("\*");

}

if(i < N)

{

columns++;

}

else

{

columns--;

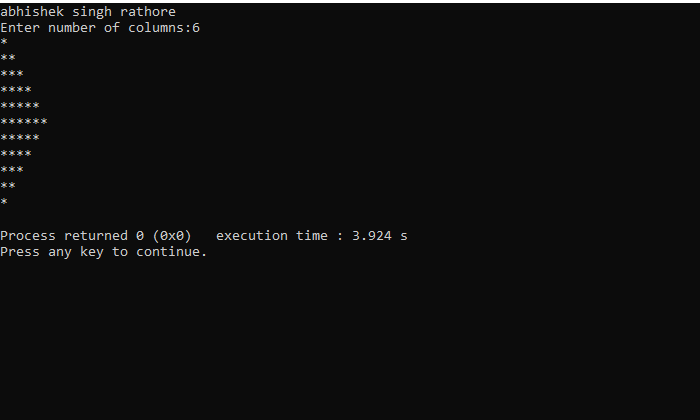
}

printf("\n");

}

return 0;

}



**6.Mirrored half diamond star pattern**

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int i, j, N;

int star, spaces;

printf(" ");

printf("\nEnter number of columns : ");

scanf("%d", &N);

spaces = N-1;

star = 1;

for(i=1; i<N\*2; i++)

{

for(j=1; j<=spaces; j++)

printf(" ");

for(j=1; j<=star; j++)

printf("\*");

printf("\n");

if(i < N)

{

star++;

spaces--;

}

else

{

star--;

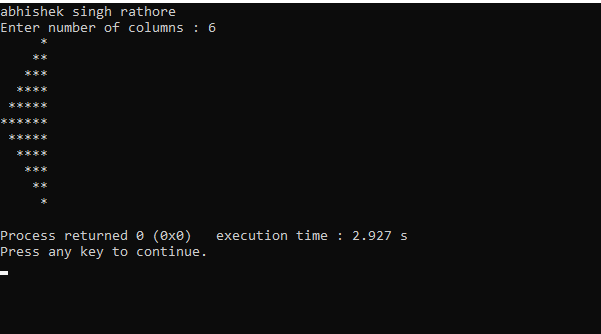
spaces++;

}

}

return 0;

}



**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()

{

printf("abhishek singh rathore");

int rows, cols, i, j;

printf(" ");

printf("\nEnter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

for(i=1; i<=rows; i++)

{

for(j=1; j<=cols; j++)

{

printf("1");

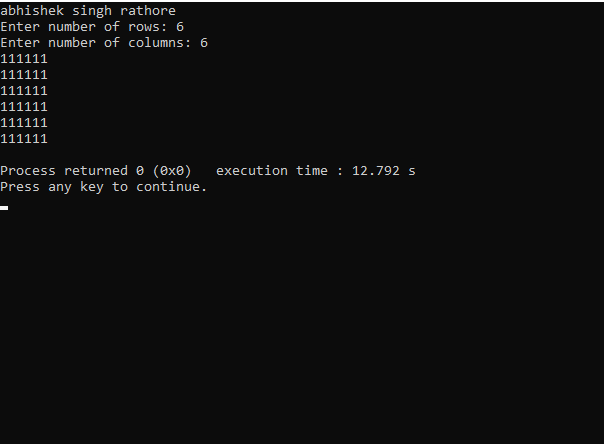
}

printf("\n");

}

return 0;

}



**Number pattern 1**

**11111**

**00000**

**11111**

**00000**

**11111**

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int rows, cols, i, j;

printf(" ");

printf("\nEnter number of rows: ");

scanf("%d", &rows);

printf("\nEnter number of columns: ");

scanf("%d", &cols);

for(i=1; i<=rows; i++)

{

for(j=1; j<=cols; j++)

{

if(i%2 == 1)

{

printf("1");

}

else

{

printf("0");

}

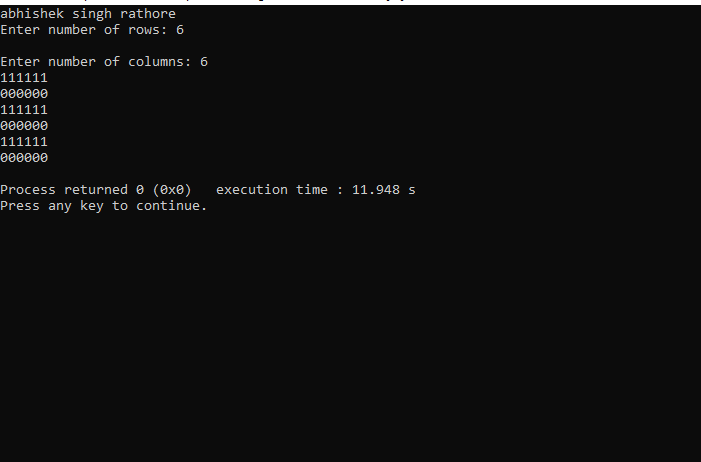
}

printf("\n");

}

return 0;

}



**Number pattern 2**

**01010**

**01010**

**01010**

**01010**

**01010**

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int rows, cols, i, j;

printf(" ");

printf("\nEnter number of rows: ");

scanf("%d", &rows);

printf("\nEnter number of columns: ");

scanf("%d", &cols);

for(i=1; i<=rows; i++)

{

for(j=1; j<=cols; j++)

{

if(j%2 == 1)

{

printf("0");

}

else

{

printf("1");

}

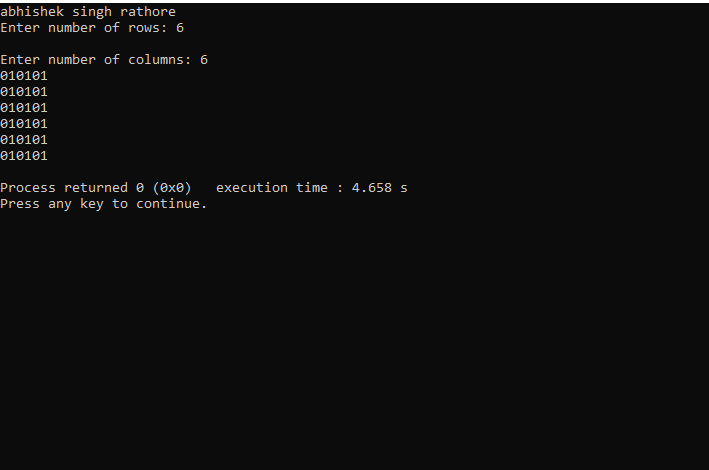
}

printf("\n");

}

return 0;

}



**Number pattern 3**

**11111**

**10001**

**10001**

**10001**

**11111**

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int rows, cols, i, j;

printf(" ");

printf("\nEnter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

for(i=1; i<=rows; i++)

{

for(j=1; j<=cols; j++)

{

if(i==1 || i==rows || j==1 || j==cols)

{

printf("1");

}

else

{

printf("0");

}

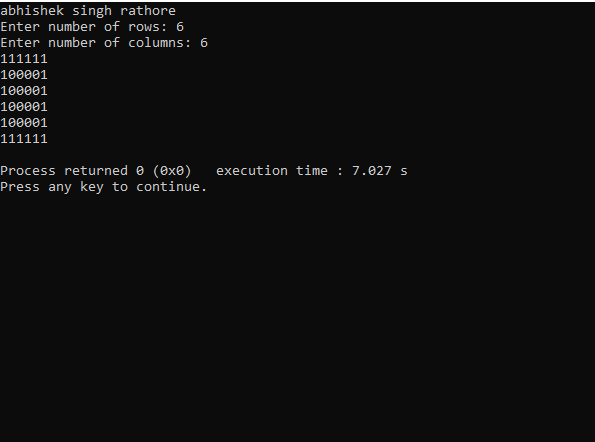
}

printf("\n");

}

return 0;

}



**Number pattern 4**

**11111**

**11111**

**11011**

**11111**

**11111**

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int rows, cols, i, j;

int centerRow, centerCol;

printf(" ");

printf("\nEnter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

centerRow = (rows + 1) / 2;

centerCol = (cols + 1) / 2;

for(i=1; i<=rows; i++)

{

for(j=1; j<=cols; j++)

{

if(centerCol == j && centerRow == i)

{

printf("0");

}

else if(cols%2 == 0 && centerCol+1 == j)

{

if(centerRow == i || (rows%2 == 0 && centerRow+1 == i))

printf("0");

else

printf("1");

}

else if(rows%2 == 0 && centerRow+1 == i)

{

if(centerCol == j || (cols%2 == 0 && centerCol+1 == j))

printf("0");

else

printf("1");

}

else

{

printf("1");

}

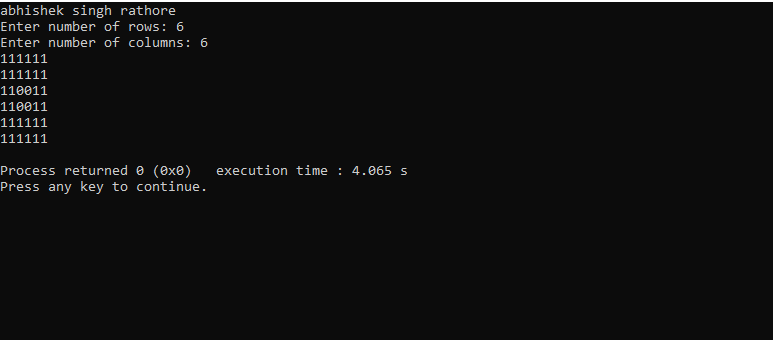
}

printf("\n");

}

return 0;

}



--Number pattern 5

10101

01010

10101

01010

10101

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int rows, cols, i, j, k;

printf(" ");

printf("\nEnter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

k = 1;

for(i=1; i<=rows; i++)

{

for(j=1; j<=cols; j++)

{

if(k == 1)

{

printf("1");

}

else

{

printf("0");

}

k \*= -1;

}

if(cols % 2 == 0)

{

k \*= -1;

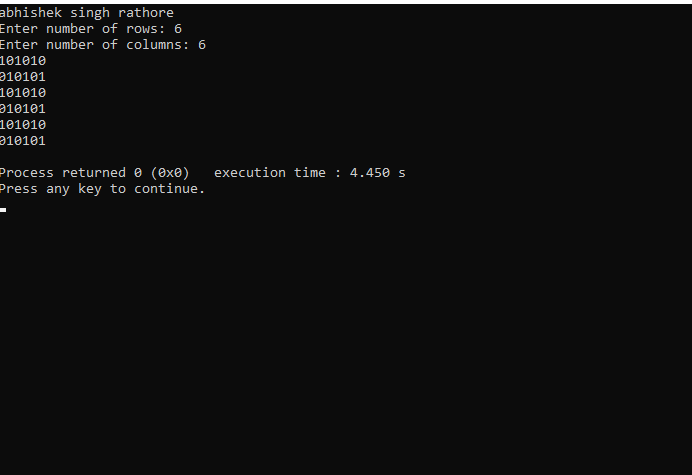
}

printf("\n");

}

return 0;

}



**If…Else Exercises**

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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int num1, num2;

printf(" ");

printf("\nEnter two numbers: ");

scanf("%d%d", &num1, &num2);

if(num1 > num2)

{

printf("%d is maximum", num1);

}

if(num2 > num1)

{

printf("%d is maximum", num2);

}

if(num1 == num2)

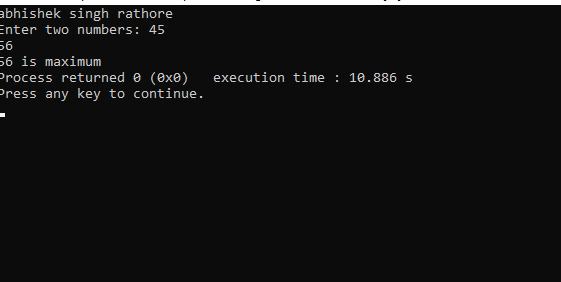
{

printf("Both are equal");

}

return 0;

}



1. **Write a C program to find maximum between three numbers.**

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int num1, num2, num3, max;

printf(" ");

printf("\nEnter three numbers: ");

scanf("%d%d%d", &num1, &num2, &num3);

if(num1 > num2)

{

if(num1 > num3)

{

max = num1;

}

else

{

max = num3;

}

}

else

{

if(num2 > num3)

{

max = num2;

}

else

{

max = num3;

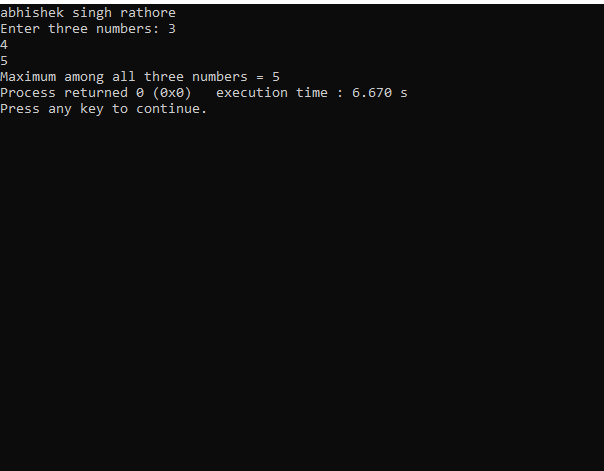
}

}

printf("Maximum among all three numbers = %d", max);

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int num;

printf(" ");

printf("\nEnter any number: ");

scanf("%d", &num);

if(num > 0)

{

printf("Number is POSITIVE");

}

else if(num < 0)

{

printf("Number is NEGATIVE");

}

else

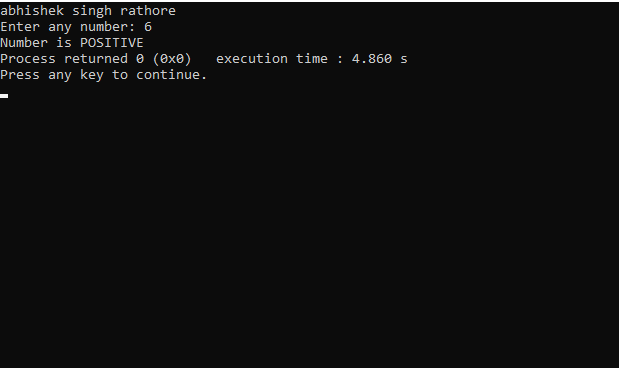
{

printf("Number is ZERO");

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int num;

printf(" ");

printf("\nEnter any number: ");

scanf("%d", &num);

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

{

printf("Number is divisible by 5 and 11");

}

else

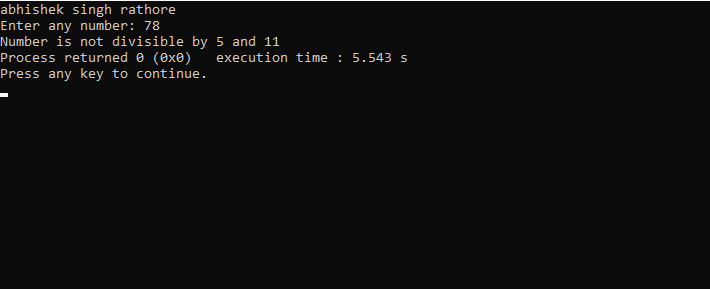
{

printf("Number is not divisible by 5 and 11");

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int num;

printf(" ");

printf("\nEnter any number to check even or odd: ");

scanf("%d", &num);

if(num % 2 == 0)

{

printf("Number is Even.");

}

else

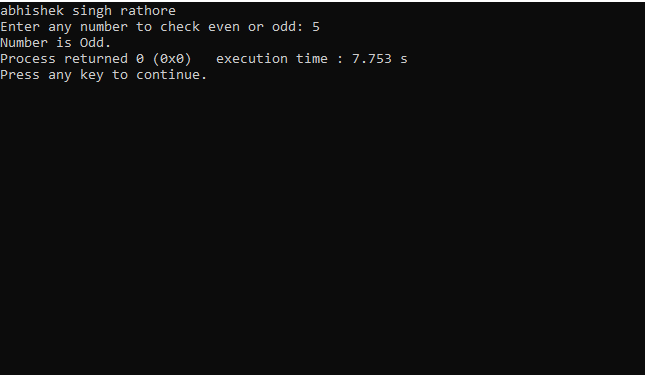
{

printf("Number is Odd.");

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore;");

int year;

printf(" ");

printf("\nEnter year : ");

scanf("%d", &year);

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

{

printf("LEAP YEAR");

}

else

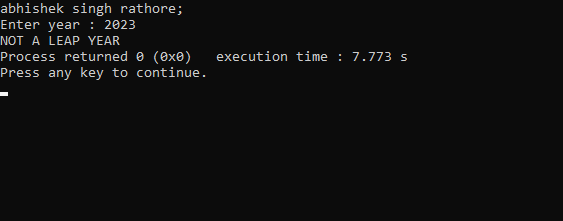
{

printf("NOT A LEAP YEAR");

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

char ch;

printf(" ");

printf("\nEnter any character: ");

scanf("%c", &ch);

if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))

{

printf("Character is an ALPHABET.");

}

else

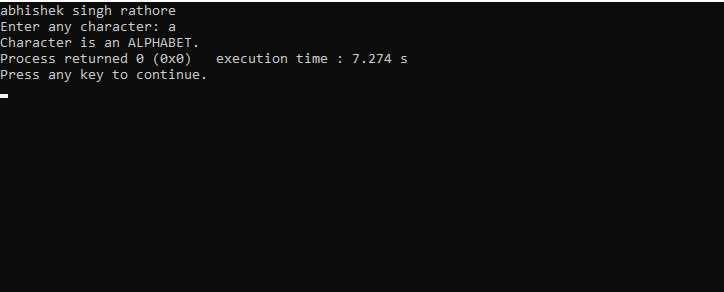
{

printf("Character is NOT ALPHABET.");

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

char ch;

printf(" ");

printf("\nEnter any character: ");

scanf("%c", &ch);

if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' ||

ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')

{

printf("'%c' is Vowel.", ch);

}

else if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))

{

printf("'%c' is Consonant.", ch);

}

else

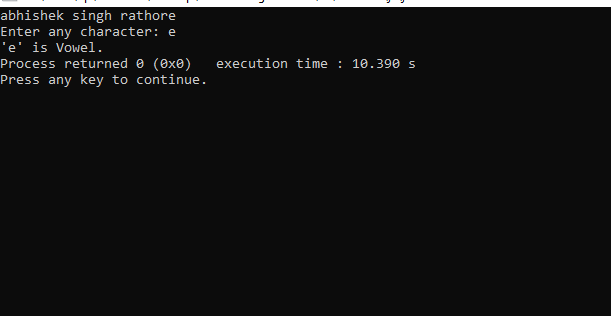
{

printf("'%c' is not an alphabet.", ch);

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

char ch;

printf(" ");

printf("\nEnter any character: ");

scanf("%c", &ch);

if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))

{

printf("'%c' is alphabet.", ch);

}

else if(ch >= '0' && ch <= '9')

{

printf("'%c' is digit.", ch);

}

else

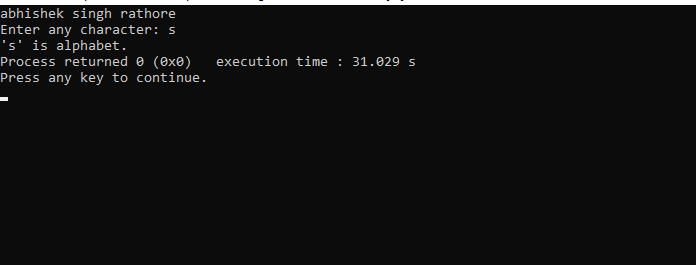
{

printf("'%c' is special character.", ch);

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

char ch;

printf(" ");

printf("\nEnter any character: ");

scanf("%c", &ch);

if(ch >= 'A' && ch <= 'Z')

{

printf("'%c' is uppercase alphabet.", ch);

}

else if(ch >= 'a' && ch <= 'z')

{

printf("'%c' is lowercase alphabet.", ch);

}

else

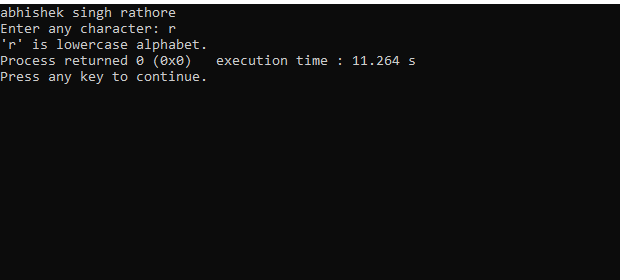
{

printf("'%c' is not an alphabet.", ch);

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int week;

printf(" ");

printf("\nEnter day number (1-7): ");

scanf("%d", &week);

if(week == 1)

{

printf("Monday");

}

else if(week == 2)

{

printf("Tuesday");

}

else if(week == 3)

{

printf("Wednesday");

}

else if(week == 4)

{

printf("Thursday");

}

else if(week == 5)

{

printf("Friday");

}

else if(week == 6)

{

printf("Saturday");

}

else if(week == 7)

{

printf("Sunday");

}

else

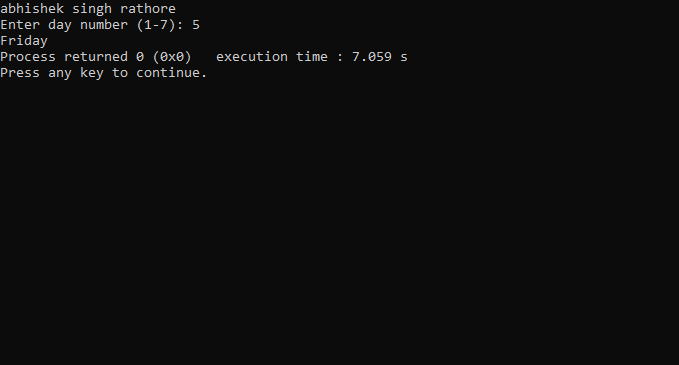
{

printf("Invalid Input! Please enter week number between 1-7.");

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int month;

printf(" ");

printf("\nEnter month number (1-12): ");

scanf("%d", &month);

if(month == 1)

{

printf("31 days");

}

else if(month == 2)

{

printf("28 or 29 days");

}

else if(month == 3)

{

printf("31 days");

}

else if(month == 4)

{

printf("30 days");

}

else if(month == 5)

{

printf("31 days");

}

else if(month == 6)

{

printf("30 days");

}

else if(month == 7)

{

printf("31 days");

}

else if(month == 8)

{

printf("31 days");

}

else if(month == 9)

{

printf("30 days");

}

else if(month == 10)

{

printf("31 days");

}

else if(month == 11)

{

printf("30 days");

}

else if(month == 12)

{

printf("31 days");

}

else

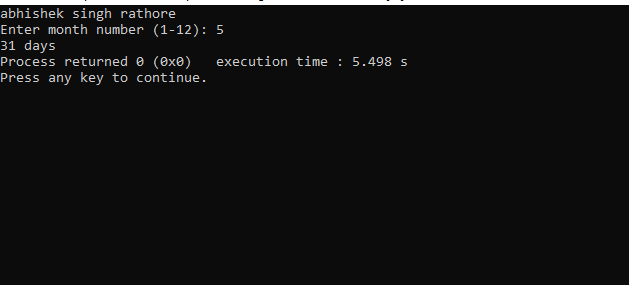
{

printf("Invalid input! Please enter month number between (1-12).");

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int amount;

int note500, note100, note50, note20, note10, note5, note2, note1;note500 = note100 = note50 = note20 = note10 = note5 = note2 = note1 = 0;

printf(" ");

printf("\nEnter amount: ");

scanf("%d", &amount);

if(amount >= 500)

{

note500 = amount/500;

amount -= note500 \* 500;

}

if(amount >= 100)

{

note100 = amount/100;

amount -= note100 \* 100;

}

if(amount >= 50)

{

note50 = amount/50;

amount -= note50 \* 50;

}

if(amount >= 20)

{

note20 = amount/20;

amount -= note20 \* 20;

}

if(amount >= 10)

{

note10 = amount/10;

amount -= note10 \* 10;

}

if(amount >= 5)

{

note5 = amount/5;

amount -= note5 \* 5;

}

if(amount >= 2)

{

note2 = amount /2;

amount -= note2 \* 2;

}

if(amount >= 1)

{

note1 = amount;

}

printf("Total number of notes = \n");

printf("500 = %d\n", note500);

printf("100 = %d\n", note100);

printf("50 = %d\n", note50);

printf("20 = %d\n", note20);

printf("10 = %d\n", note10);

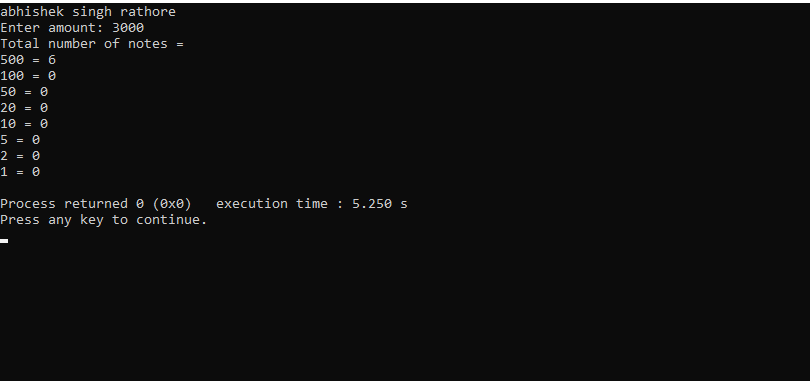
printf("5 = %d\n", note5);

printf("2 = %d\n", note2);

printf("1 = %d\n", note1);

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int angle1, angle2, angle3, sum;

printf(" ");

printf("\nEnter three angles of triangle: \n");

scanf("%d%d%d", &angle1, &angle2, &angle3);

sum = angle1 + angle2 + angle3;

if(sum == 180 && angle1 > 0 && angle2 > 0 && angle3 > 0)

{

printf("Triangle is valid.");

}

else

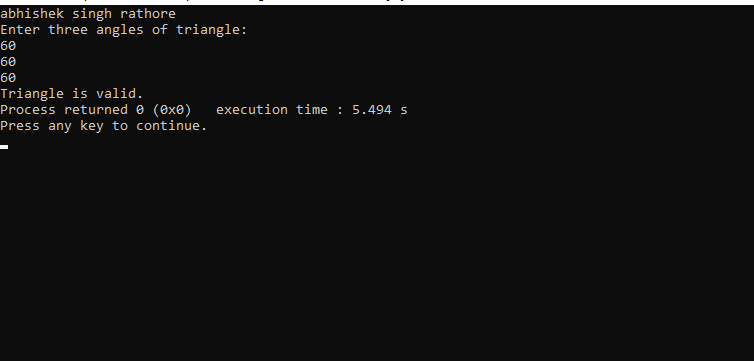
{

printf("Triangle is not valid.");

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int side1, side2, side3;

printf(" ");

printf("\nEnter three sides of triangle: \n");

scanf("%d%d%d", &side1, &side2, &side3);

if((side1 + side2) > side3)

{

if((side2 + side3) > side1)

{

if((side1 + side3) > side2)

{

printf("Triangle is valid.");

}

else

{

printf("Triangle is not valid.");

}

}

else

{

printf("Triangle is not valid.");

}

}

else

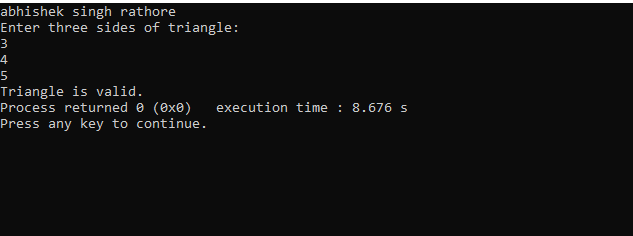
{

printf("Triangle is not valid.");

}

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int side1, side2, side3;

printf(" ");

printf("\nEnter three sides of triangle: ");

scanf("%d%d%d", &side1, &side2, &side3);

if(side1==side2 && side2==side3)

{

printf("Equilateral triangle.");

}

else if(side1==side2 || side1==side3 || side2==side3)

{

printf("Isosceles triangle.");

}

else

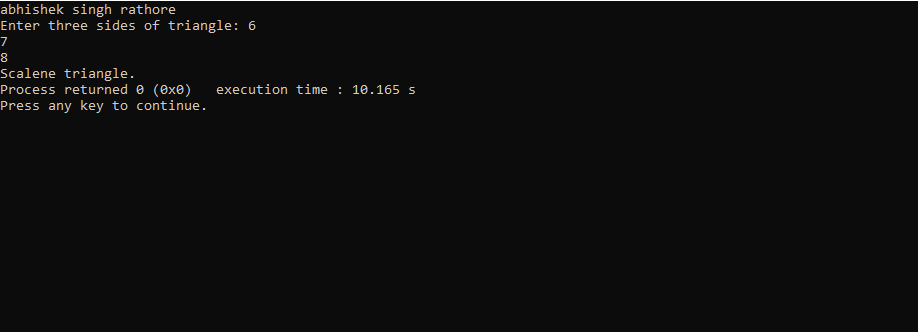
{

printf("Scalene triangle.");

}

return 0;

}



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

#include <stdio.h>

#include <math.h> /

int main()

{

printf("abhishek singh rathore");

float a, b, c;

float root1, root2, imaginary;

float discriminant;

printf(" ");

printf("\nEnter values of a, b, c of quadratic equation (aX^2 + bX + c): ");

scanf("%f%f%f", &a, &b, &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 and real roots exists: %.2f and %.2f", root1, root2);

}

else if(discriminant == 0)

{

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

printf("Two equal and real roots exists: %.2f and %.2f", root1, root2);

}

else if(discriminant < 0)

{

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

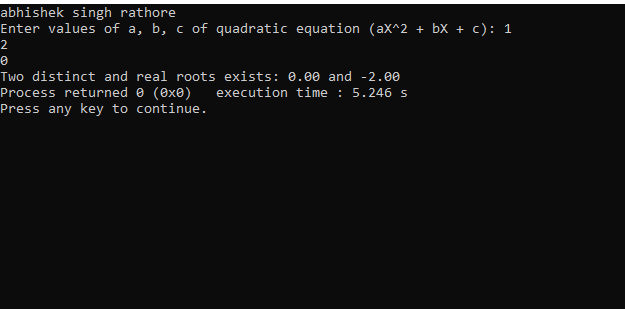
imaginary = sqrt(-discriminant) / (2 \* a);

printf("Two distinct complex roots exists: %.2f + i%.2f and %.2f - i%.2f", root1, imaginary, root2, imaginary);

}

return 0;

}



1. **Write a C program to calculate profit or loss.**

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int cp,sp, amt;

printf(" ");

printf("\nEnter cost price: ");

scanf("%d", &cp);

printf("Enter selling price: ");

scanf("%d", &sp);

if(sp > cp)

{

amt = sp - cp;

printf("Profit = %d", amt);

}

else if(cp > sp)

{

amt = cp - sp;

printf("Loss = %d", amt);

}

else

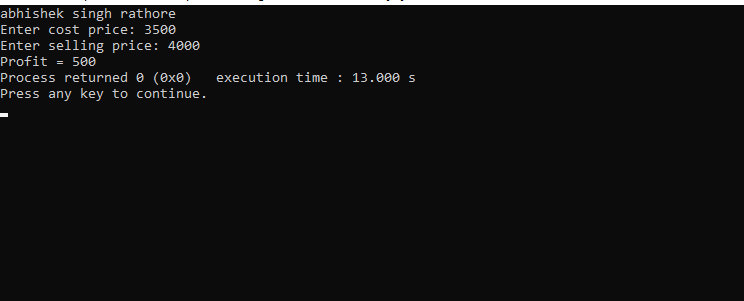
{

printf("No Profit No Loss.");

}

return 0;

}



**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()

{

printf("abhishek singh rathore");

int phy, chem, bio, math, comp;

float per;

printf(" ");

printf("\nEnter five subjects marks: ");

scanf("%d%d%d%d%d", &phy, &chem, &bio, &math, &comp);

per = (phy + chem + bio + math + comp) / 5.0;

printf("Percentage = %.2f\n", per);

if(per >= 90)

{

printf("Grade A");

}

else if(per >= 80)

{

printf("Grade B");

}

else if(per >= 70)

{

printf("Grade C");

}

else if(per >= 60)

{

printf("Grade D");

}

else if(per >= 40)

{

printf("Grade E");

}

else

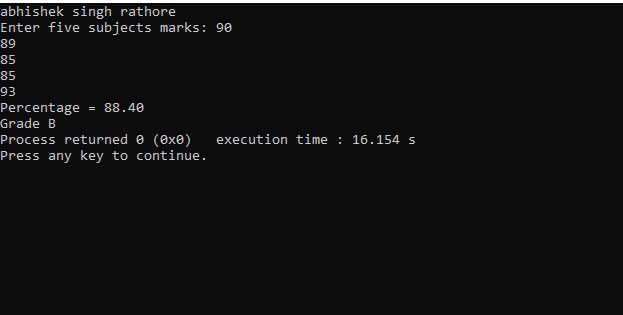
{

printf("Grade F");

}

return 0;

}



**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()

{

printf("abhishek singh rathore");

float basic, gross, da, hra;

printf(" ");

printf("\nEnter basic salary of an employee: ");

scanf("%f", &basic);

if(basic <= 10000)

{

da = basic \* 0.8;

hra = basic \* 0.2;

}

else if(basic <= 20000)

{

da = basic \* 0.9;

hra = basic \* 0.25;

}

else

{

da = basic \* 0.95;

hra = basic \* 0.3;

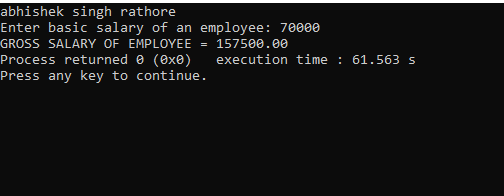
}

gross = basic + hra + da;

printf("GROSS SALARY OF EMPLOYEE = %.2f", gross);

return 0;

}



**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()

{

printf("abhishek singh rathore");

int unit;

float amt, total\_amt, sur\_charge;

printf(" ");

printf("\nEnter total units consumed: ");

scanf("%d", &unit);

if(unit <= 50)

{

amt = unit \* 0.50;

}

else if(unit <= 150)

{

amt = 25 + ((unit-50) \* 0.75);

}

else if(unit <= 250)

{

amt = 100 + ((unit-150) \* 1.20);

}

else

{

amt = 220 + ((unit-250) \* 1.50);

}

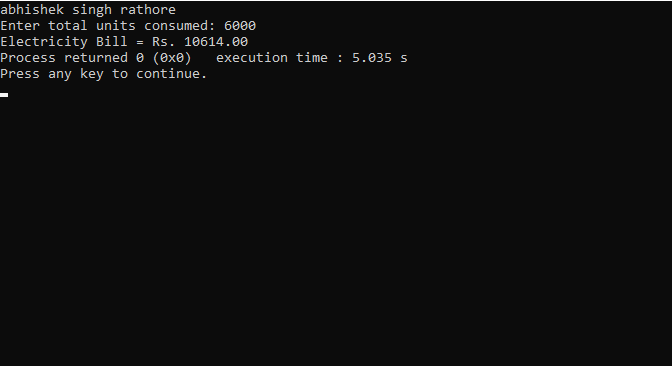
sur\_charge = amt \* 0.20;

total\_amt = amt + sur\_charge;

printf("Electricity Bill = Rs. %.2f", total\_amt);

return 0;

}



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

#include <stdio.h>

int main()

{

printf("abhishek singh rathore");

int days, years, weeks;

printf(" ");

days = 1329;

years = days/365;

weeks = (days % 365)/7;

days = days- ((years\*365) + (weeks\*7));

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

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

printf("Days: %d \n", days);

return 0;

}

