 GRAPHIC ERA

DEEMED TO BE UNIVERSITY

INTRODUCTION TO C PROGRAMMING

BATCH: (2023-2026)

B.C.A. 1ST YAER

SUBMITTED BY. SUBMITTED TO.

KHEM RAJ JOSHI MS. POOJA CHAHAR

ST.ID. 2316012166 CSIT , GEU

1. WAP to find the largest number using the logical AND operator.

#include <stdio.h>

int main() {

int n;

printf("khem raj joshi\n");

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

scanf("%d", &n);

if (n <= 0) {

printf("Invalid input. Please enter a positive number of elements.\n");

return 1;

}

int largest;

int first = 1;

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

int num;

printf("Enter number %d: ", i + 1);

scanf("%d", &num);

if (first || (num > largest)) {

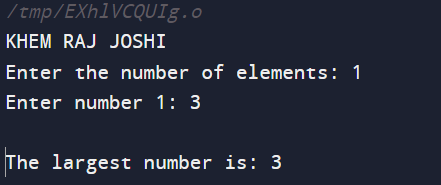
largest = num;

first = 0; // Set the flag to 0 after the first number is entered

}

}

printf("The largest number is: %d\n", largest);

 return 0;

}

2. WAP to validate whether the username and password entered by the user are correct or not using the predefined username and password.

#include <stdio.h>

#include <string.h>

int main() {

char correctUsername[] = "KHEM RAJ JOSHI";

char correctPassword[] = "123456789";

char enteredUsername[50];

char enteredPassword[50];

printf("abhishek sharma \n");

printf("Enter username: ");

scanf("%s", enteredUsername);

printf("Enter password: ");

scanf("%s", enteredPassword);

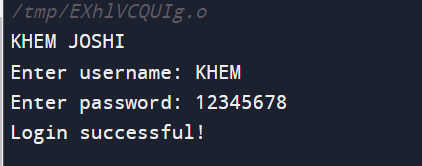
if (strcmp(enteredUsername, correctUsername) == 0 && strcmp(enteredPassword, correctPassword) == 0) {

printf("Login successful!\n");

} else {

printf("Login failed. Please check your username and password.\n");

}

 return 0;

}

3. WAP to input the positive number from the user to perform the left shift operator.

#include <stdio.h>

int main() {

int num, shift;

{

printf("khem raj joshi \n");

printf("Enter a positive integer: ");

scanf("%d", &num);

if (num <= 0) {

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

}

} while (num <= 0);

printf("Enter the number of positions to shift left: ");

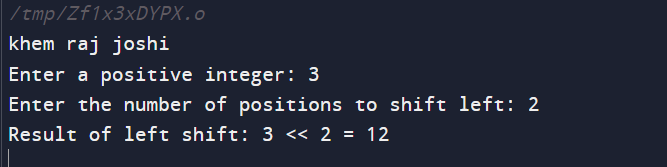
scanf("%d", &shift);

int result = num << shift;

printf("Result of left shift: %d << %d = %d\n", num, shift, result);

return 0;

}



4. WAP to input the positive number from the user to perform the right shift operator.

#include <stdio.h>

int main() {

int num, shift;

{

printf("\n");

printf("Enter a positive integer: ");

scanf("%d", &num);

if (num <= 0) {

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

}

} while (num <= 0);

printf("Enter the number of positions to shift right: ");

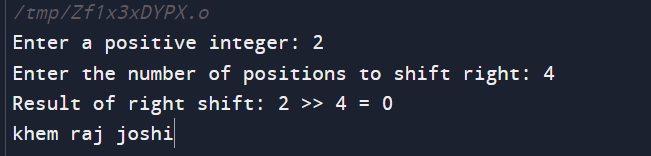
scanf("%d", &shift);

int result = num >> shift;

printf("Result of right shift: %d >> %d = %d\n", num, shift, result);

return 0;

}



5. WAP to perform the pre-increment and pre-decrement operator on two integers and printf both the original value and updated value.

#include <stdio.h>

int main() {

int num1, num2;

printf("khem raj joshi \n");

printf("Enter the first integer: ");

scanf("%d", &num1);

printf("Enter the second integer: ");

scanf("%d", &num2);

int preIncNum1 = ++num1;

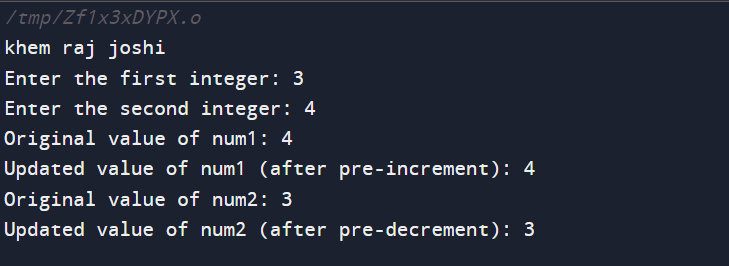
int preDecNum2 = --num2;

printf("Original value of num1: %d\n", num1);

printf("Updated value of num1 (after pre-increment): %d\n", preIncNum1);

printf("Original value of num2: %d\n", num2);

printf("Updated value of num2 (after pre-decrement): %d\n", preDecNum2);

return 0;

}

6. WAP to perform the post-increment and post-decrement operator on two integers and printf both the original value and updated value.

#include <stdio.h>

int main() {

int num1, num2;

printf("khem raj joshi \n");

printf("Enter the first integer: ");

scanf("%d", &num1);

printf("Enter the second integer: ");

scanf("%d", &num2);

int postIncNum1 = num1++;

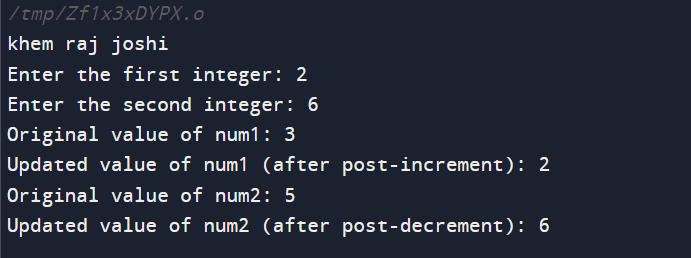
int postDecNum2 = num2--;

printf("Original value of num1: %d\n", num1);

printf("Updated value of num1 (after post-increment): %d\n", postIncNum1);

printf("Original value of num2: %d\n", num2);

printf("Updated value of num2 (after post-decrement): %d\n", postDecNum2);

return 0;

}

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

#include <stdio.h>

int main() {

int num;

printf("khem raj joshi \n");

printf("Enter an integer: ");

scanf("%d", &num);

if (num % 9 == 0 || num % 7 == 0) {

printf("%d is divisible by 9 or 7.\n", num);

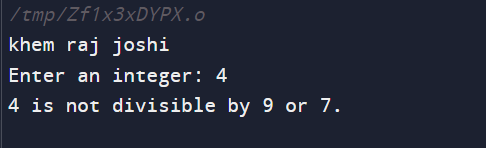
} else {

printf("%d is not divisible by 9 or 7.\n", num);

}

return 0;

}



8. WAP to identify gender in a single character and printf full gender (e.g.: if the input is 'M' or 'm' – it should printf "Male").

#include <stdio.h>

int main() {

char gender;

printf("khem raj joshi \n");

printf("Enter gender (M/F): ");

scanf(" %c", &gender);

switch (gender) {

case 'M':

case 'm':

printf("Male\n");

break;

case 'F':

case 'f':

printf("Female\n");

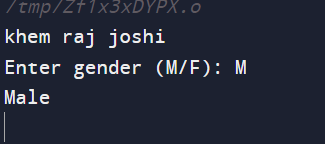
break;

default:

printf("Invalid gender input\n");

}

return 0;

}

8. WAP for Hello World or this is my first c program.

#include <stdio.h>

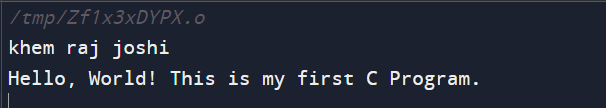
int main() {

printf("khem raj joshi \n");

printf("Hello, World! This is my first C Program.\n");

return 0;

}



1. WAP to add two numbers.

#include <stdio.h>

int main() {

int num1, num2, sum;

printf("khem raj joshi\n");

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

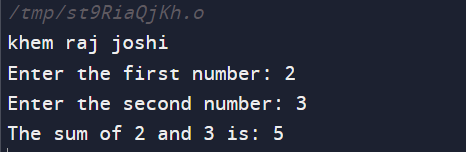
scanf("%d", &num2);

sum = num1 + num2;

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

return 0;

}



10. WAP to find the area of a circle.

#include <stdio.h>

#include <math.h>

int main() {

double radius, area;

printf("khem raj joshi\n");

printf("Enter the radius of the circle: ");

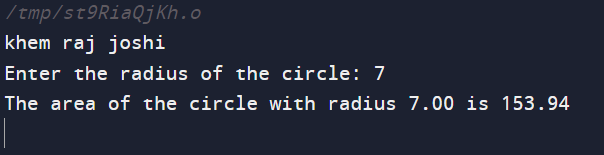
scanf("%lf", &radius);

area = M\_PI \* pow(radius, 2);

printf("The area of the circle with radius %.2lf is %.2lf\n", radius, area);

return 0;

}



11. WAP to divide two numbers.

#include <stdio.h>

int main() {

double num1, num2, result;

printf("khem raj joshi\n");

printf("Enter the first number: ");

scanf("%lf", &num1);

printf("Enter the second number: ");

scanf("%lf", &num2);

(division by zero is undefined)

if (num2 != 0) {

result = num1 / num2;

printf("The result of %.2lf divided by %.2lf is: %.2lf\n", num1, num2, result);

} else {

printf("Error: Division by zero is not allowed.\n");

}

return 0;

}

12. WAP to printf ASCII value.

#include <stdio.h>

int main() {

char ch;

printf("khem raj joshi\n");

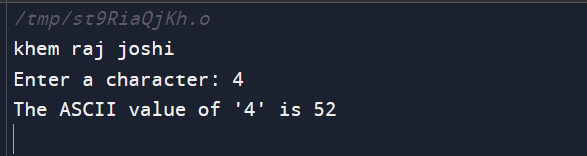
printf("Enter a character: ");

scanf(" %c", &ch);

printf("The ASCII value of '%c' is %d\n", ch, ch);

return 0;

}



13. WAP to multiply floating point numbers.

#include <stdio.h>

int main() {

double num1, num2, result;

printf("khem raj joshi\n");

printf("Enter the first floating-point number: ");

scanf("%lf", &num1);

printf("Enter the second floating-point number: ");

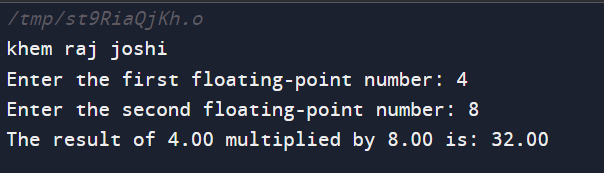
scanf("%lf", &num2);

result = num1 \* num2;

printf("The result of %.2lf multiplied by %.2lf is: %.2lf\n", num1, num2, result);

return 0;

}



14. WAP to add two numbers.

#include <stdio.h>

int main() {

int num1, num2, sum;

printf("khem raj joshi\n");

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

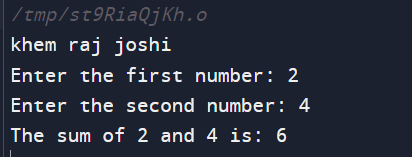
scanf("%d", &num2);

sum = num1 + num2;

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

return 0;

}



15. WAP to find the area of a circle.

#include <stdio.h>

#include <math.h>

int main() {

double radius, area;

printf("khem raj joshi \n");

printf("Enter the radius of the circle: ");

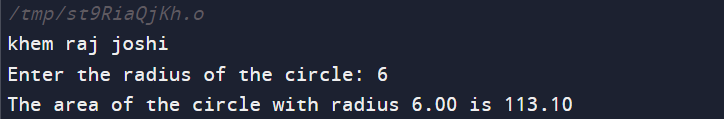
scanf("%lf", &radius);

area = M\_PI \* pow(radius, 2);

printf("The area of the circle with radius %.2lf is %.2lf\n", radius, area);

return 0;

}



16. WAP to SWAP two variables number by using a third variable.

#include <stdio.h>

int main() {

int num1, num2, temp;

printf("khem raj joshi \n");

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

scanf("%d", &num2);

temp = num1;

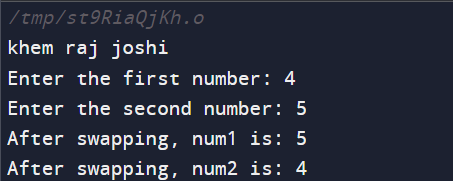
num1 = num2;

num2 = temp;

printf("After swapping, num1 is: %d\n", num1);

printf("After swapping, num2 is: %d\n", num2);

return 0;

}

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

int main() {

int num1, num2, num3;

printf("KHEM RAJ JOSHI\n");

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

scanf("%d", &num2);

printf("Enter the third number: ");

scanf("%d", &num3);

num1 = num1 + num2 + num3;

num2 = num1 - (num2 + num3);

num3 = num1 - (num2 + num3);

num1 = num1 - (num2 + num3);

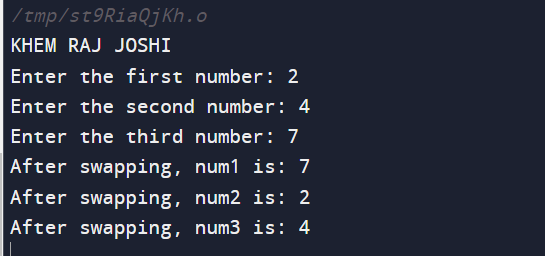
printf("After swapping, num1 is: %d\n", num1);

printf("After swapping, num2 is: %d\n", num2);

printf("After swapping, num3 is: %d\n", num3);

return 0;

}



18. WAP to find the area of a rectangle.

#include <stdio.h>

int main() {

double length, width, area;

printf("khem raj joshi\n");

printf("Enter the length of the rectangle: ");

scanf("%lf", &length);

printf("Enter the width of the rectangle: ");

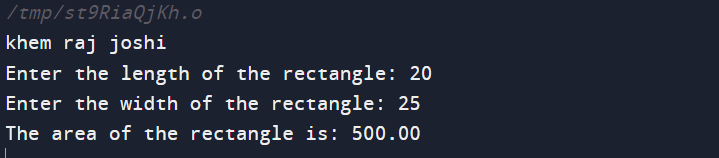
scanf("%lf", &width);

area = length \* width;

printf("The area of the rectangle is: %.2lf\n", area);

return 0;

}



20. WAP to find the area of a rectangle.

#include <stdio.h>

int main() {

double length, width, area;

printf("khem raj joshi \n");

printf("Enter the length of the rectangle: ");

scanf("%lf", &length);

printf("Enter the width of the rectangle: ");

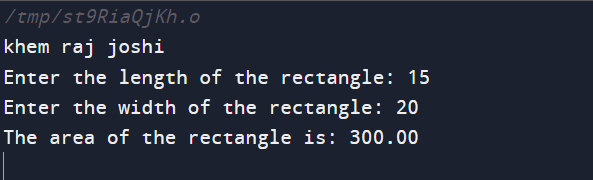
scanf("%lf", &width);

area = length \* width;

printf("The area of the rectangle is: %.2lf\n", area);

return 0;

}



21. WAP to find the area of the right angle triangle, isosceles triangle, and any triangle with three sides.

#include <stdio.h>

#include <math.h>

double areaOfRightAngledTriangle(double base, double height) {

return 0.5 \* base \* height;

}

double areaOfIsoscelesTriangle(double base, double equalSide) {

double height = sqrt(pow(equalSide, 2) - pow(base / 2, 2));

return 0.5 \* base \* height;

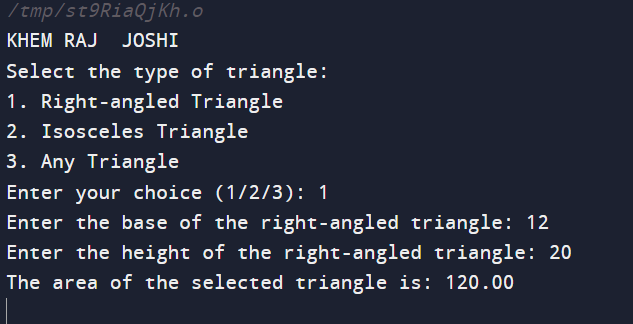
}

double area Of Any Triangle(double a, double b, double c) {

double s = (a + b + c) / 2; // Semi-perimeter

return sqrt(s \* (s - a) \* (s - b) \* (s - c));

}



22. WAP to find the area and volume of a cube.

#include <stdio.h>

int main() {

double side, area, volume;

printf("khem raj joshi \n");

printf("Enter the length of one side of the cube: ");

scanf("%lf", &side);

area = 6 \* side \* side;

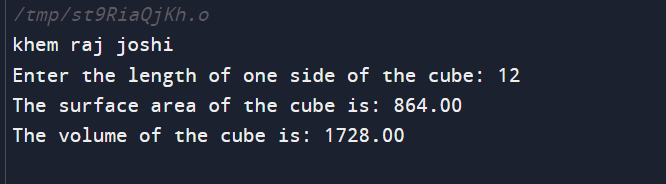
volume = side \* side \* side;

printf("The surface area of the cube is: %.2lf\n", area);

printf("The volume of the cube is: %.2lf\n", volume);

return 0;

}



23. WAP to find the area and volume of the cuboid.

#include <stdio.h>

int main() {

double length, width, height, surface area, and volume;

printf("khem raj joshi \n");

printf("Enter the length of the cuboid: ");

scanf("%lf", &length);

printf("Enter the width of the cuboid: ");

scanf("%lf", &width);

printf("Enter the height of the cuboid: ");

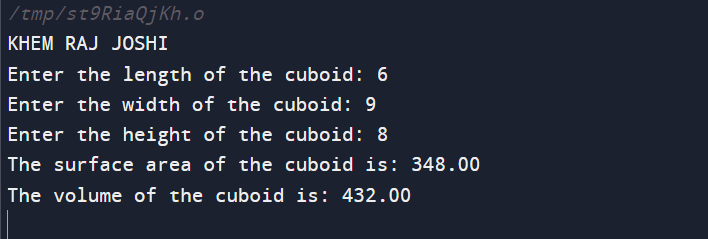
scanf("%lf", &height);

surfaceArea = 2 \* (length \* width + width \* height + height \* length);

volume = length \* width \* height;

printf("The surface area of the cuboid is: %.2lf\n", surfaceArea);

printf("The volume of the cuboid is: %.2lf\n", volume);

return 0;

}

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

to the given condition:

For first 50 units Rs. 0.50/unit

For next 100 units Rs. 0.75/unit

For next 100 units Rs. 1.20/unit

For unit above 250 Rs. 1.50/unit

An additional surcharge of 20% is added to the bill

#include <stdio.h>

int main() {

float units, bill;

printf("Enter the number of units consumed: ");

scanf("%f", &units);

if (units <= 50) {

bill = units \* 0.50;

} else if (units <= 150) {

bill = 50 \* 0.50 + (units - 50) \* 0.75;

} else if (units <= 250) {

bill = 50 \* 0.50 + 100 \* 0.75 + (units - 150) \* 1.20;

} else {

bill = 50 \* 0.50 + 100 \* 0.75 + 100 \* 1.20 + (units - 250) \* 1.50;

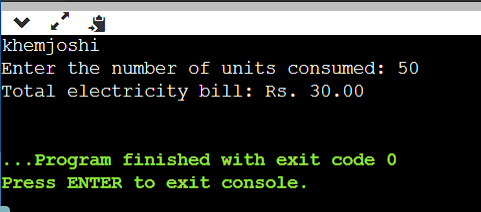
}

bill += 0.2 \* bill;

printf("Total electricity bill: Rs. %.2f\n", bill);

return 0;

output:



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

to following:

Basic Salary <= 10000 : HRA = 20%, DA = 80%

Basic Salary <= 20000 : HRA = 25%, DA = 90%

Basic Salary > 20000 : HRA = 30%, DA = 95%

#include<stdio.h>

int main() {

float basic\_salary, hra, da, gross\_salary;

// Input basic salary

printf("Enter the basic salary: ");

scanf("%f", &basic\_salary);

// Calculate HRA and DA based on conditions

if (basic\_salary <= 10000) {

hra = 0.2 \* basic\_salary;

da = 0.8 \* basic\_salary;

} else if (basic\_salary <= 20000) {

hra = 0.25 \* basic\_salary;

da = 0.9 \* basic\_salary;

} else {

hra = 0.3 \* basic\_salary;

da = 0.95 \* basic\_salary;

}

// Calculate gross salary

gross\_salary = basic\_salary + hra + da;

// Print the results

printf("Basic Salary: %.2f\n", basic\_salary);

printf("HRA: %.2f\n", hra);

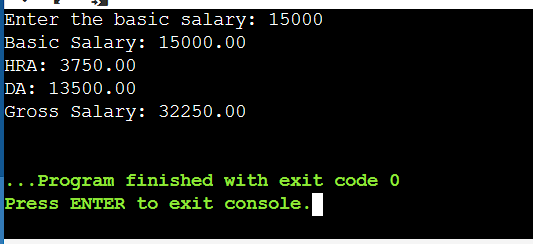
printf("DA: %.2f\n", da);

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

return 0;

}

Output :



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

and Computer. Calculate percentage and grade according to following:

Percentage >= 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 total\_marks, percentage;

char grade;

printf("Enter marks in Physics: ");

scanf("%f", &physics);

printf("Enter marks in Chemistry: ");

scanf("%f", &chemistry);

printf("Enter marks in Biology: ");

scanf("%f", &biology);

printf("Enter marks in Mathematics: ");

scanf("%f", &mathematics);

printf("Enter marks in Computer: ");

scanf("%f", &computer);

total\_marks = physics + chemistry + biology + mathematics + computer;

percentage = (total\_marks / 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';

}

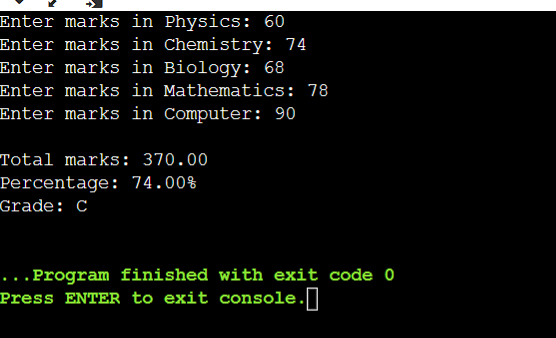
printf("\nTotal marks: %.2f\n", total\_marks);

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

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

return 0;

}

Output :

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

#include <stdio.h>

int main() {

char c;

printf("Enter a character: ");

scanf("%c", &c);

if (c >= 'A' && c <= 'Z') {

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

}

else if (c >= 'a' && c <= 'z') {

printf("%c is a lowercase alphabet.\n", c);

}

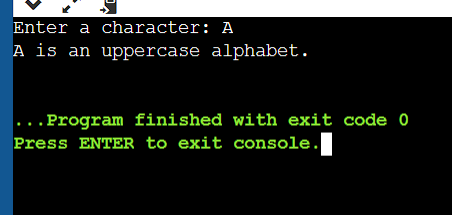
else {

printf("%c is not an alphabet.\n", c);

}

return 0;

}



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("Sunday\n");

break;

case 2:

printf("Monday\n");

break;

case 3:

printf("Tuesday\n");

break;

case 4:

printf("Wednesday\n");

break;

case 5:

printf("Thursday\n");

break;

case 6:

printf("Friday\n");

break;

case 7:

printf("Saturday\n");

break;

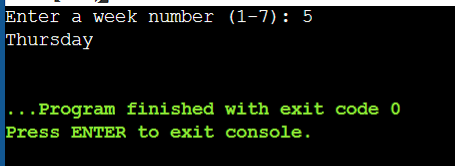
default:

printf("Invalid input. Please enter a number between 1 and 7.\n");

}

return 0;

}



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

#include <stdio.h>

int main() {

int month;

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

scanf("%d", &month);

if (month < 1 || month > 12) {

printf("Invalid month number. Please enter a number between 1 and 12.\n");

return 1; // Return an error code

}

int days;

switch(month) {

case 2: // February

days = 28;

break;

case 4: case 6: case 9: case 11: // April, June, September, November

days = 30;

break;

default: // All other months

days = 31;

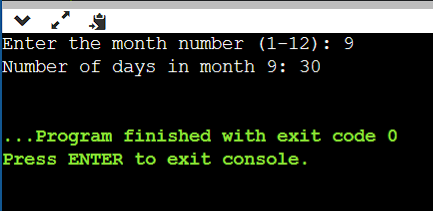
break;

}

printf("Number of days in month %d: %d\n", month, days);

return 0;

}



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

#include <stdio.h>

int main() {

int amount;

int notes[6] = {2000, 500, 100, 50, 20, 10}; // Denominations of notes

// Input the amount

printf("Enter the amount: ");

scanf("%d", &amount);

// Initialize a counter for each type of note

int note\_count[6] = {0};

// Iterate through the notes and count how many of each are needed

for (int i = 0; i < 6; i++) {

note\_count[i] = amount / notes[i];

amount = amount % notes[i];

}

// Display the results

for (int i = 0; i < 6; i++) {

if (note\_count[i] > 0) {

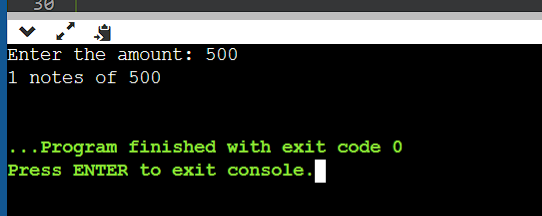
printf("%d notes of %d\n", note\_count[i], notes[i]);

}

}

return 0;

}



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

#include <stdio.h>

int main() {

int angle1, angle2, angle3;

printf("Enter angle 1: ");

scanf("%d", &angle1);

printf("Enter angle 2: ");

scanf("%d", &angle2);

printf("Enter angle 3: ");

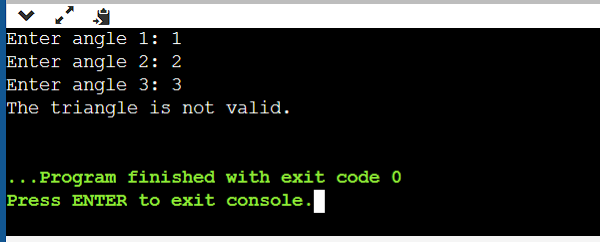
scanf("%d", &angle3);

if (angle1 + angle2 + angle3 == 180 && angle1 > 0 && angle2 > 0 && angle3 > 0) {

printf("The triangle is valid.\n");

} else {

printf("The triangle is not valid.\n");

 }

return 0;

}

17. 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 lengths of three sides of the triangle: ");

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

if (side1 + side2 > side3 && side1 + side3 > side2 && side2 + side3 > side1) {

printf("The sides form a valid triangle.\n");

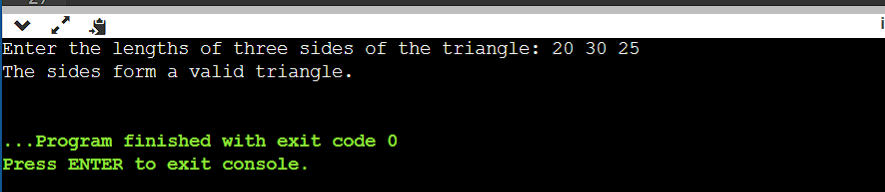
} else {

printf("The sides do not form a valid triangle.\n");

}

return 0;

}



18. 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 lengths of three sides of the triangle: ");

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

if (side1 + side2 > side3 && side1 + side3 > side2 && side2 + side3 > side1)

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

printf("It is an equilateral triangle.\n");

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

printf("It is an isosceles triangle.\n");

} else {

printf("It is a scalene triangle.\n");

}

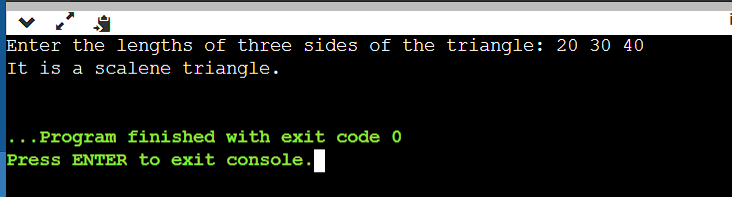
} else {

printf("The given sides do not form a valid triangle.\n");

}

return 0;

}



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

#include <stdio.h>

#include <math.h>

int main() {

double a, b, c;

double discriminant, root1, root2;

printf("Enter coefficients (a, b, c): ");

scanf("%lf %lf %lf", &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("Roots are real and different.\n");

printf("Root 1 = %.2lf\n", root1);

printf("Root 2 = %.2lf\n", root2);

}

else if (discriminant == 0) {

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

printf("Roots are real and same.\n");

printf("Root 1 = Root 2 = %.2lf\n", root1);

}

else {

double realPart = -b / (2\*a);

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

printf("Roots are complex and different.\n");

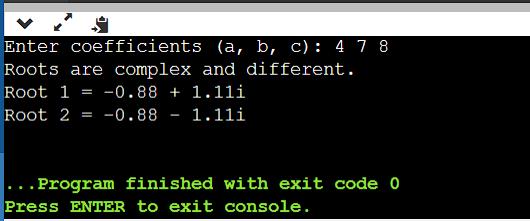
printf("Root 1 = %.2lf + %.2lfi\n", realPart, imaginaryPart);

printf("Root 2 = %.2lf - %.2lfi\n", realPart, imaginaryPart);

}

return 0;

}



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

#include <stdio.h>

int main()

{

int days, years, weeks;

days = 1329;

#include <stdio.h>

int main() {

int days, years, weeks;

// Input the number of days

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

scanf("%d", &days);

// Convert days into years, weeks, and days

years = days / 365;

days = days % 365;

weeks = days / 7;

days = days % 7;

// Output the result

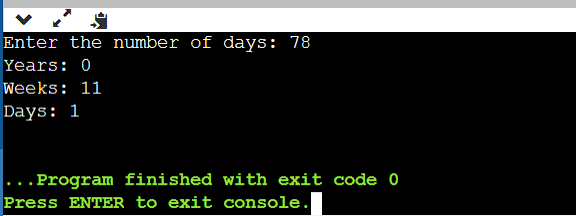
printf("Years: %d\n", years);

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

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

return 0;

}



#include <stdio.h>

int main() {

int days, years, weeks;

days = 1329;

// Converts days to years, weeks, and days

years = days / 365;

int remainingDays = days % 365;

weeks = remainingDays / 7;

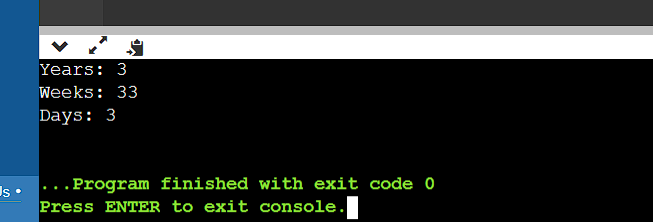
int remainingDays2 = remainingDays % 7;

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

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

printf("Days: %d\n", remainingDays2);

return 0;

}

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

#include <stdio.h>

int main() {

float cost\_price, selling\_price, profit\_loss;

// Get cost price and selling price from the user

printf("Enter cost price: ");

scanf("%f", &cost\_price);

printf("Enter selling price: ");

scanf("%f", &selling\_price);

// Calculate profit or loss

profit\_loss = selling\_price - cost\_price;

// Determine if it's a profit or loss and display the result

if (profit\_loss > 0) {

printf("Profit: %.2f\n", profit\_loss);

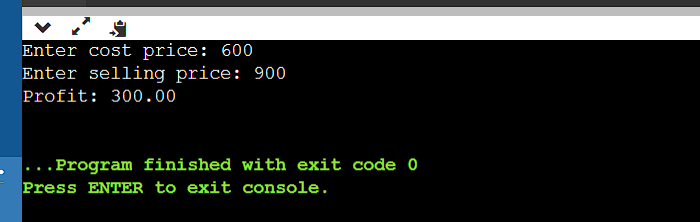
} else if (profit\_loss < 0) {

printf("Loss: %.2f\n", -profit\_loss);

} else {

printf("No profit, no loss.\n");

}



return 0;

}

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

#include <stdio.h>

int main() {

int n;

// Prompt user for input

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

scanf("%d", &n);

// Check if n is positive

if (n <= 0) {

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

return 1; // Return an error code

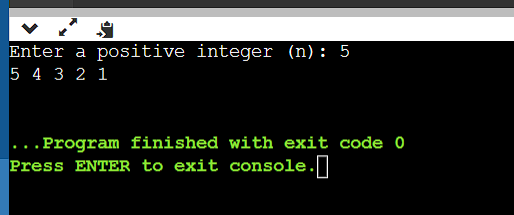
}

// Print natural numbers in reverse order

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

printf("%d ", i);

}

 printf("\n");

return 0;

}

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

#include <stdio.h>

int main() {

char alphabet;

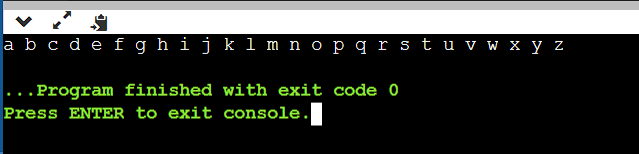
for (alphabet = 'a'; alphabet <= 'z'; alphabet++) {

printf("%c ", alphabet);

}

return 0;

}



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

#include <stdio.h>

int main() {

int n, i;

// Ask the user for 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");

return 1; // Return an error code

}

// Loop from 1 to n and print the natural numbers

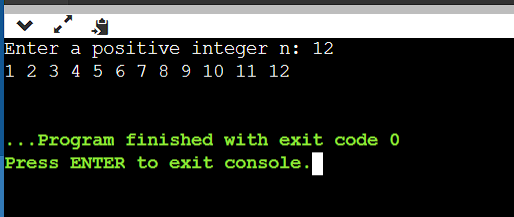
for (i = 1; i <= n; i++) {

printf("%d ", i);

}

printf("\n");

return 0;

}

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

#include<stdio.h>

int main() {

int i;

printf("\nEven numbers from 1 to 100 :\n");

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

if (i % 2 != 0) {

continue;

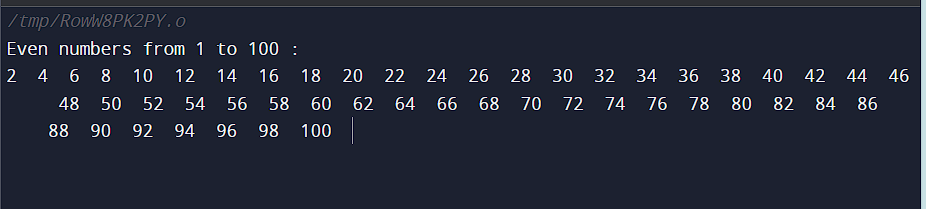
}

printf("%d ", i);

}

return 0;

}



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

#include<stdio.h>

int main() {

int i;

printf("\nOdd numbers from 1 to 100 :\n");

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

if (i % 2 == 0) {

continue;

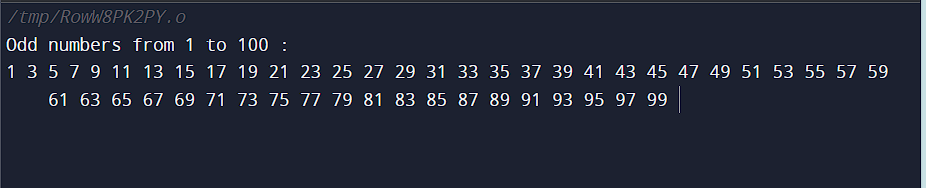
}

printf("%d ", i);

}

return 0;

}



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

#include<stdio.h>

int main()

{

int i, n, sum = 0;

printf("\nEnter any number: ");

scanf("%d", &n);

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

printf("%d ", i);

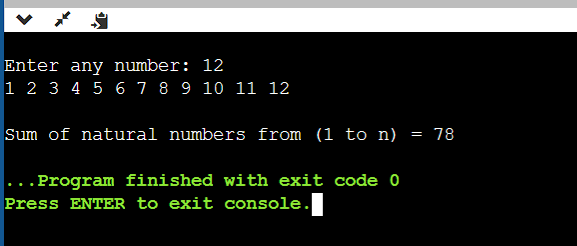
sum = sum + i;

}

printf("\n\nSum of natural numbers from (1 to n) = %d", sum);

return 0;

}



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

#include<stdio.h>

int main()

{

int i, n, sum = 0;

printf("\nEnter any number: ");

scanf("%d", &n);

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

if (i % 2 != 0) {

continue;

}

printf("%d ", i);

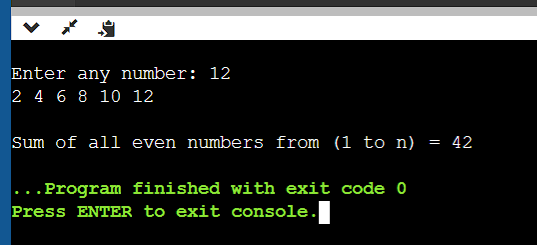
sum = sum + i;

}

printf("\n\nSum of all even numbers from (1 to n) = %d", sum);

return 0;

}



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

#include<stdio.h>

int main()

{

int i, n, sum = 0;

printf("\nEnter any number: ");

scanf("%d", &n);

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

if (i % 2 == 0) {

continue;

}

printf("%d ", i);

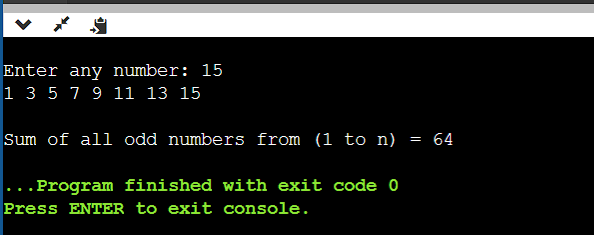
sum = sum + i;

}

printf("\n\nSum of all odd numbers from (1 to n) = %d", sum);

return 0;

}



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

#include<stdio.h>

int main()

{

int i, n, product;

printf("\nEnter any number: ");

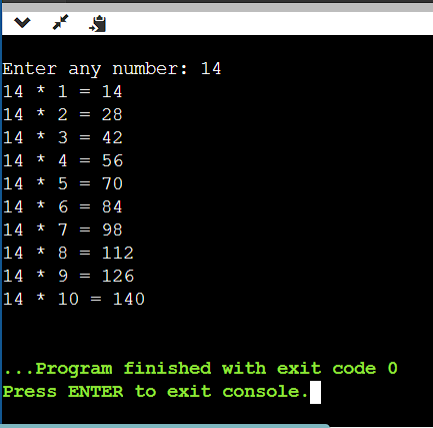
scanf("%d", &n);

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

product = n \* i;

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

}

 return 0;

}

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

#include<stdio.h>

int main()

{

int n, count = 0;

printf("\nEnter any number: ");

scanf("%d", &n);

while (n > 0) {

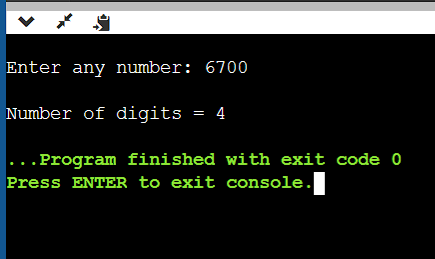
n = n / 10;

count++;

}

printf("\nNumber of digits = %d", count);

return 0;

}

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

#include <stdio.h>

int main() {

int number, firstDigit, lastDigit;

printf("Enter a number: ");

scanf("%d", &number);

lastDigit = number % 10;

while (number >= 10) {

number /= 10;

}

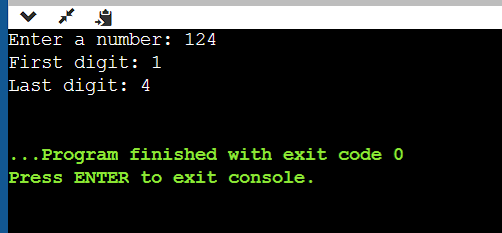
firstDigit = number;

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() {

int num, firstDigit, lastDigit, sum;

printf("Enter a number: ");

scanf("%d", &num);

lastDigit = num % 10;

while (num >= 10) {

num /= 10;

}

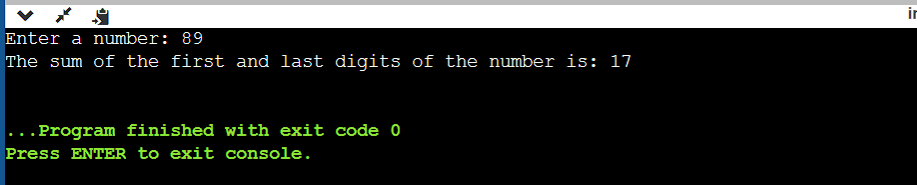
firstDigit = num;

sum = firstDigit + lastDigit;

printf("The sum of the first and last digits of the number is: %d\n", sum);

return 0;

}



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

#include <stdio.h>

int main() {

int number, originalNumber, firstDigit, lastDigit, temp;

printf("Enter a number: ");

scanf("%d", &number);

originalNumber = number;

// Get the last digit

lastDigit = number % 10;

// Find the number of digits in the given number

while (number >= 10) {

number /= 10;

}

// The first digit is now the remaining number

firstDigit = number;

// Swap the first and last digits

temp = firstDigit;

firstDigit = lastDigit;

lastDigit = temp;

// Reconstruct the new number

int swappedNumber = 0;

number = originalNumber;

while (number >= 10) {

swappedNumber = swappedNumber \* 10 + (number % 10);

number /= 10;

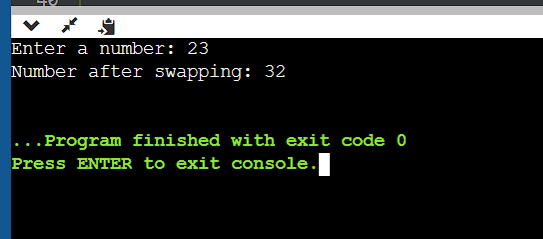
}

swappedNumber = swappedNumber \* 10 + number;

printf("Number after swapping: %d\n", swappedNumber);

return 0;

}



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

#include <stdio.h>

int main() {

int num, sum = 0, digit;

printf("Enter an integer: ");

scanf("%d", &num);

while (num > 0) {

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

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

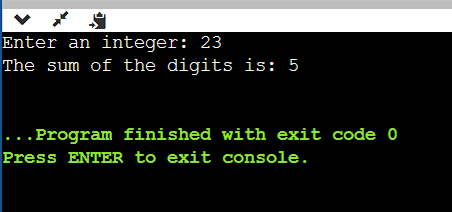
num = num / 10; // Remove the last digit

}

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

return 0;

}



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

#include <stdio.h>

int main() {

int num, digit, product = 1;

/\* Input number from the user \*/

printf("Enter a number: ");

scanf("%d", &num);

while (num != 0) {

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

product \*= digit; // Multiply it with the product

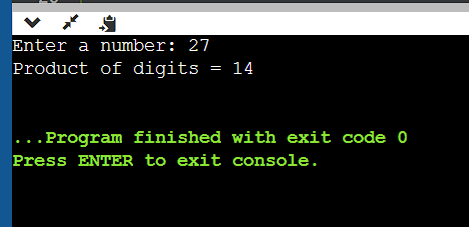
num = num / 10; // Remove the last digit

}

printf("Product of digits = %d\n", product);

return 0;

}



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

#include <stdio.h>

int main() {

int num, reversed = 0;

printf("Enter a number: ");

scanf("%d", &num);

while (num != 0) {

int digit = num % 10;

reversed = reversed \* 10 + digit;

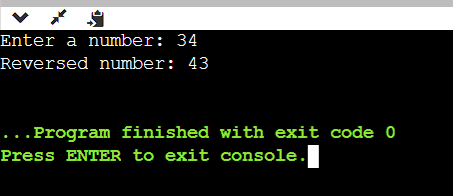
num /= 10;

}

printf("Reversed number: %d\n", reversed);

return 0;

}



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

#include <stdio.h>

int main() {

int num, originalNum, reversed = 0;

printf("Enter a number: ");

scanf("%d", &num);

originalNum = num; // Store the original number

// Reverse the number

while (num != 0) {

int digit = num % 10;

reversed = reversed \* 10 + digit;

num /= 10;

}

// Check if the reversed number is the same as the original number

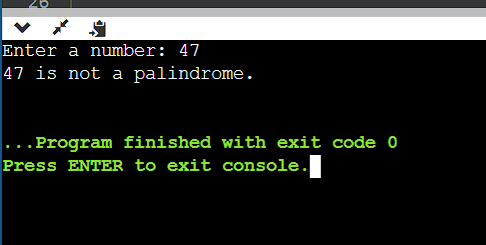
if (originalNum == reversed) {

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() {

int num, digit, count;

int frequency[10] = {0}; // Initialize an array to store the frequency of each digit

printf("Enter an integer: ");

scanf("%d", &num);

while (num != 0) {

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

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

num /= 10; // Remove the last digit

}

printf("Digit Frequency:\n");

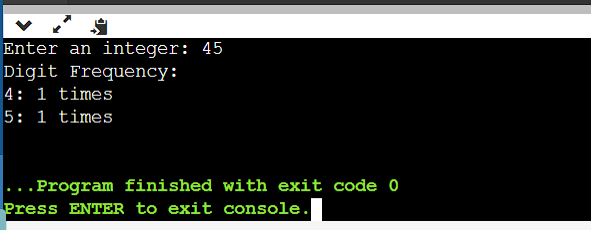
for (digit = 0; digit <= 9; digit++) {

if (frequency[digit] > 0) {

printf("%d: %d times\n", digit, frequency[digit]);

}

}



return 0;

}

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

#include <stdio.h>

// Function to print a number in words for single-digit numbers

void printDigitInWords(int digit) {

switch (digit) {

case 0:

printf("Zero ");

break;

case 1:

printf("One ");

break;

case 2:

printf("Two ");

break;

case 3:

printf("Three ");

break;

case 4:

printf("Four ");

break;

case 5:

printf("Five ");

break;

case 6:

printf("Six ");

break;

case 7:

printf("Seven ");

break;

case 8:

printf("Eight ");

break;

case 9:

printf("Nine ");

break;

}

}

int main() {

int num, digit, reversed = 0;

printf("Enter a number: ");

scanf("%d", &num);

if (num == 0) {

printf("Zero\n");

return 0;

}

// Reverse the number to print it correctly

while (num != 0) {

digit = num % 10;

reversed = reversed \* 10 + digit;

num /= 10;

}

// Print the number in words

while (reversed != 0) {

digit = reversed % 10;

printDigitInWords(digit);

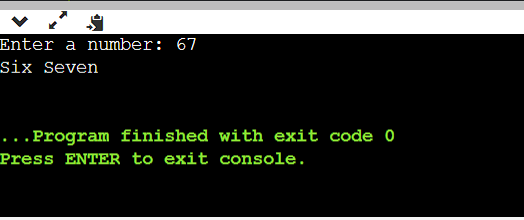
reversed /= 10;

}

printf("\n");

return 0;

}



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

#include <stdio.h>

int main() {

printf("ABHISHEK SHARMA");

int i;

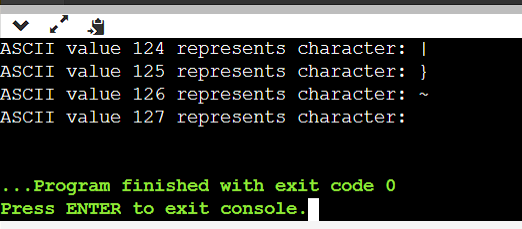
for (i = 0; i < 128; i++) {

printf("ASCII value %d represents character: %c\n", i, i);

}

return 0;

}



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

#include <stdio.h>

int main() {

double base, exponent, result = 1;

// 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++) {

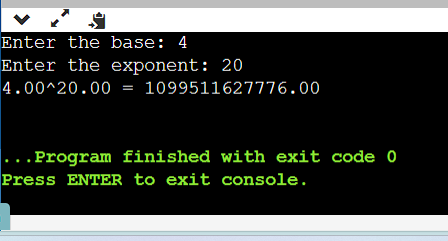
result \*= base;

}

// Print 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() {

int number;

// Input the number

printf("Enter a positive integer: ");

scanf("%d", &number);

if (number <= 0) {

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

return 1; // Exit with an error code

}

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

// Use a for loop to find and print factors

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

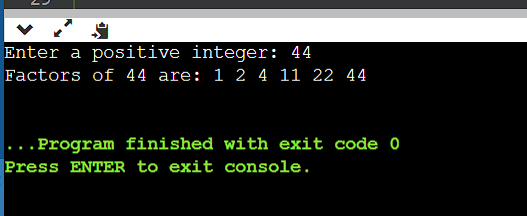
if (number % i == 0) {

printf("%d ", i);

}

}

printf("\n");

 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 findGCD(int a, int b) {

if (b == 0) {

return a;

} else {

return findGCD(b, a % b);

}

}

int main() {

int num1, num2;

printf("Enter two numbers: ");

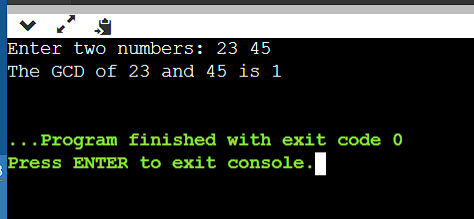
scanf("%d %d", &num1, &num2);

int gcd = findGCD(num1, num2);

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

return 0;

}



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

#include <stdio.h>

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

int findGCD(int a, int b) {

if (b == 0) {

return a;

} else {

return findGCD(b, a % b);

}

}

// Function to find the LCM of two numbers

int findLCM(int a, int b) {

int gcd = findGCD(a, b);

int lcm = (a \* b) / gcd;

return lcm;

}

int main() {

int num1, num2;

printf("Enter two numbers: ");

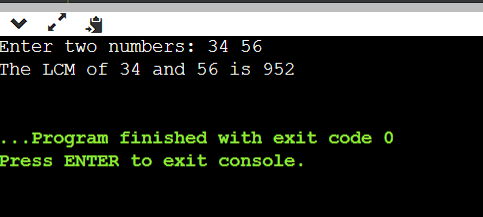
scanf("%d %d", &num1, &num2);

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>

#include <stdbool.h>

// Function to check if a number is prime

bool isPrime(int n) {

if (n <= 1) {

return false; // 0 and 1 are not prime numbers

}

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

if (n % i == 0) {

return false; // n is divisible by i, so it's not prime

}

}

return true; // If no divisors are found, it's a prime number

}

int main() {

int num;

printf("Enter a number: ");

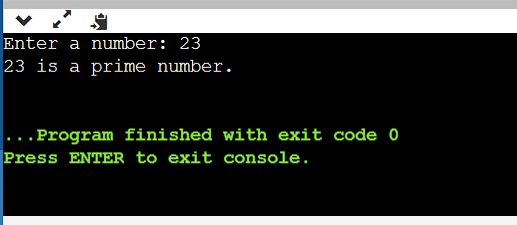
scanf("%d", &num);

if (isPrime(num)) {

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

} else {

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

 }

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() {

int n;

printf("Enter the value of 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);

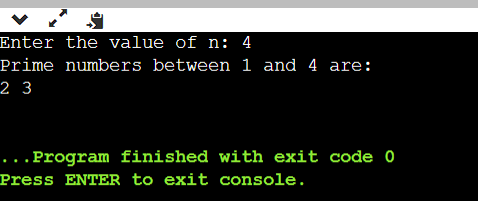
}

}

printf("\n");

return 0;

}



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

#include <stdio.h>

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() {

int n;

printf("Enter the value of n: ");

scanf("%d", &n);

int sum = 0;

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

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

if (isPrime(i)) {

printf("%d ", i);

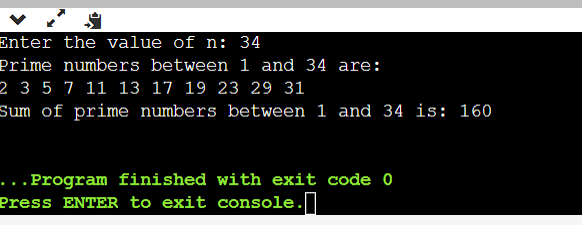
sum += i;

}

}

printf("\nSum of prime numbers between 1 and %d is: %d\n", n, sum);

return 0;

}

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

#include <stdio.h>

// Function to find and print all prime factors of a number

void primeFactors(int n) {

// Print the number of 2s that divide n

while (n % 2 == 0) {

printf("2 ");

n = n / 2;

}

// n must be odd at this point, so a skip of 2 ( i = i + 2) can be used

for (int i = 3; i \* i <= n; i = i + 2) {

// While i divides n, print i and divide n

while (n % i == 0) {

printf("%d ", i);

n = n / i;

}

}

// If n is a prime greater than 2

if (n > 2) {

printf("%d ", n);

}

}

int main() {

int n;

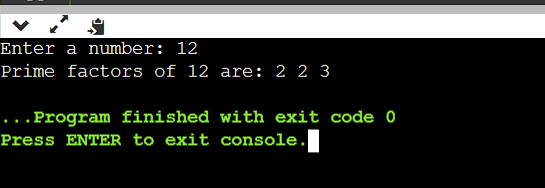
printf("Enter a number: ");

scanf("%d", &n);

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

primeFactors(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;

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; // It's an Armstrong number

else

return 0; // It's not an Armstrong number

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

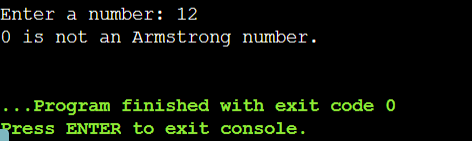
if (isArmstrong(num))

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

else

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

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, 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);

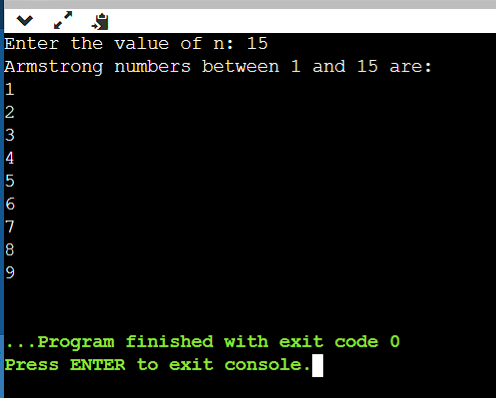
printf("Armstrong numbers between 1 and %d are:\n", n);

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

if (isArmstrong(i))

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

}



return 0;

}

55. 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;

}

}

return (sum == num);

}

int main() {

int n;

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

scanf("%d", &n);

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

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

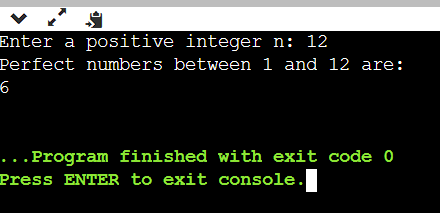
if (isPerfect(i)) {

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

}

}

return 0;

}

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

#include <stdio.h>

// Function to calculate the factorial of a number

int factorial(int num) {

int fact = 1;

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

fact \*= i;

}

return fact;

}

// Function to check if a number is a strong number

int isStrongNumber(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 a positive integer: ");

scanf("%d", &n);

if (isStrongNumber(n)) {

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

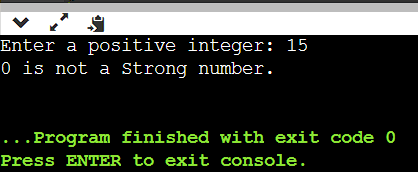
} else {

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

}

return 0;

}



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

#include <stdio.h>

// Function to calculate the factorial of a number

int factorial(int num) {

int fact = 1;

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

fact \*= i;

}

return fact;

}

// Function to check if a number is a strong number

int isStrongNumber(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 a positive integer n: ");

scanf("%d", &n);

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

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

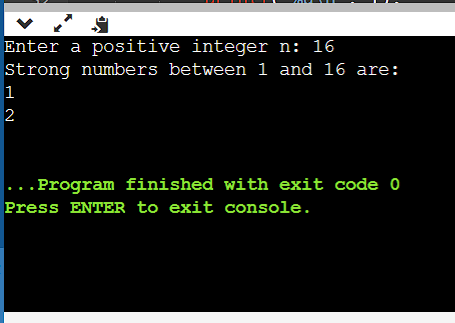
if (isStrongNumber(i)) {

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;

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

scanf("%d", &n);

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

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

if (i == 1) {

printf("%d, ", first);

} else if (i == 2) {

printf("%d, ", second);

} else {

next = first + second;

printf("%d, ", next);

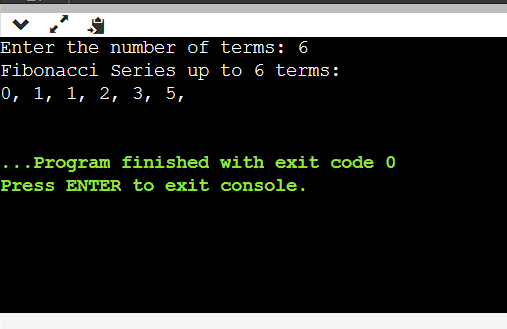
first = second;

second = 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]; // Assuming a maximum of 32-bit binary number

int length, i;

printf("Enter a binary number: ");

scanf("%s", binary);

// Find the length of the binary number

length = strlen(binary);

// Perform one's complement

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

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

binary[i] = '1';

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

binary[i] = '0';

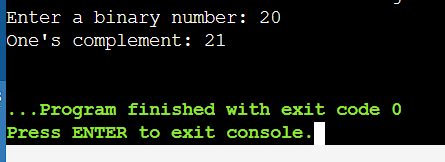
}

}

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

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to reverse a binary string

void reverseString(char str[]) {

int length = strlen(str);

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

char temp = str[i];

str[i] = str[length - i - 1];

str[length - i - 1] = temp;

}

}

// Function to add 1 to a binary string

void addOne(char binary[]) {

int length = strlen(binary);

int carry = 1;

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

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

binary[i] = '1';

carry = 0;

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

binary[i] = '0';

}

}

}

int main() {

char binary[32]; // Assuming a maximum of 32-bit binary number

printf("Enter a binary number: ");

scanf("%s", binary);

// Reverse the binary string

reverseString(binary);

// Add 1 to the reversed binary string

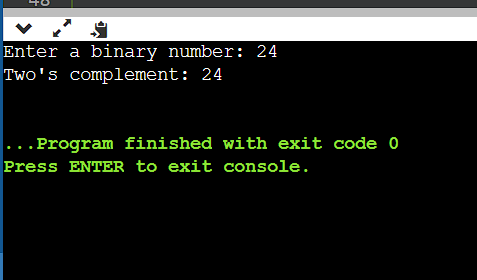
addOne(binary);

// Reverse the result to get the two's complement

reverseString(binary);

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 <string.h>

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

void binaryToOctal(char binary[]) {

int length = strlen(binary);

// Pad the binary number with leading zeros if needed to make the length a multiple of 3

int padding = (3 - (length % 3)) % 3;

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

printf("0");

}

// Iterate through the binary number in groups of 3 and convert to octal

for (int i = padding; i < length; i += 3) {

int octalDigit = (binary[i] - '0') \* 4 + (binary[i + 1] - '0') \* 2 + (binary[i + 2] - '0');

printf("%d", octalDigit);

}

printf("\n");

}

int main() {

char binary[32]; // Assuming a maximum of 32-bit binary number

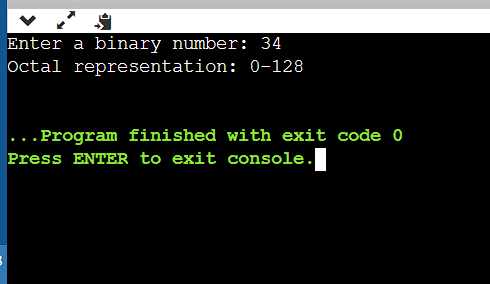
printf("Enter a binary number: ");

scanf("%s", binary);

printf("Octal representation: ");

binaryToOctal(binary);

return 0;

}

62. Write a C program to convert Binary to Decimal number system.#include <stdio.h>

#include <string.h>

// Function to convert a binary number to decimal

int binaryToDecimal(char binary[]) {

int length = strlen(binary);

int decimal = 0;

int base = 1;

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

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

decimal += base;

}

base \*= 2;

}

return decimal;

}

int main() {

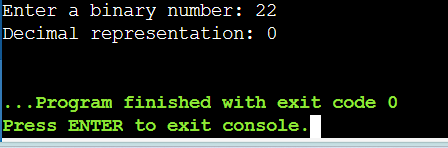
char binary[32]; // Assuming a maximum of 32-bit binary number

printf("Enter a binary number: ");

scanf("%s", binary);

int decimal = binaryToDecimal(binary);

printf("Decimal representation: %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 binary number to hexadecimal

void binaryToHexadecimal(char binary[]) {

int length = strlen(binary);

// Pad the binary number with leading zeros if needed to make the length a multiple of 4

int padding = (4 - (length % 4)) % 4;

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

printf("0");

}

// Iterate through the binary number in groups of 4 and convert to hexadecimal

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

int hexDigit = 0;

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

hexDigit = (hexDigit << 1) | (binary[i + j] - '0');

}

if (hexDigit < 10) {

printf("%d", hexDigit);

} else {

printf("%c", 'A' + hexDigit - 10);

}

}

printf("\n");

}

int main() {

char binary[32]; // Assuming a maximum of 32-bit binary number

printf("Enter a binary number: ");

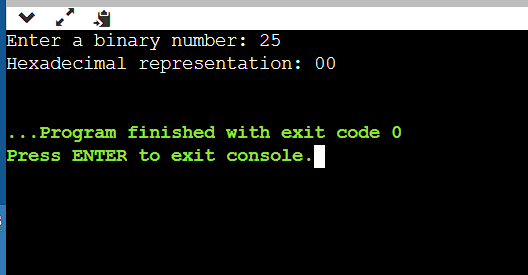
scanf("%s", binary);

printf("Hexadecimal representation: ");

binaryToHexadecimal(binary);

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 a binary string

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 "Invalid";

}

}

int main() {

char octal[32]; // Assuming a maximum of 32-bit octal number

char binary[128]; // To store the binary equivalent

printf("Enter an octal number: ");

scanf("%s", octal);

int length = strlen(octal);

int binaryIndex = 0;

// Convert each octal digit to binary and concatenate

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

char \*binaryDigit = octalToBinary(octal[i]);

// Skip the first digit if it results in leading zeros

if (i == 0 && binaryDigit[0] == '0') {

binaryDigit += 1;

}

// Copy the binary digits to the binary string

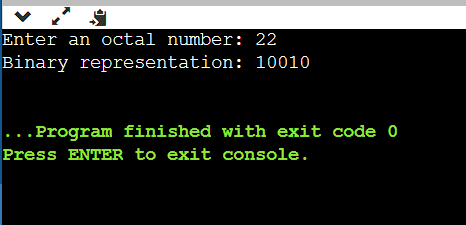
strcpy(binary + binaryIndex, binaryDigit);

binaryIndex += strlen(binaryDigit);

}

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

return 0;

}

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

#include <stdio.h>

#include <math.h>

int octalToDecimal(char octal[]) {

int length = 0;

while (octal[length] != '\0') {

length++;

}

int decimal = 0;

int base = 1;

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

if (octal[i] < '0' || octal[i] > '7') {

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

return -1;

}

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

decimal += octalDigit \* base;

base \*= 8;

}

return decimal;

}

int main() {

char octal[32]; // Assuming a maximum of 32-bit octal number

printf("Enter an octal number: ");

scanf("%s", octal);

int decimal = octalToDecimal(octal);

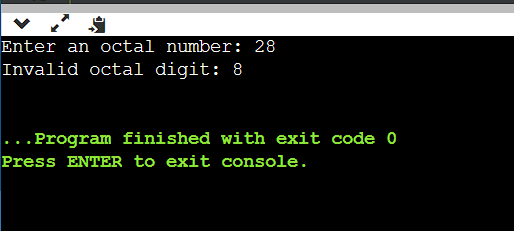
if (decimal != -1) {

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

}

return 0;

}



67. Write a C program to convert Decimal to Binary

number system.

#include <stdio.h>

void decimalToBinary(int decimal) {

if (decimal == 0) {

printf("Binary: 0\n");

return;

}

int binary[32];

int i = 0;

while (decimal > 0) {

binary[i] = decimal % 2;

decimal /= 2;

i++;

}

printf("Binary: ");

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

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

}

printf("\n");

}

int main() {

int decimal;

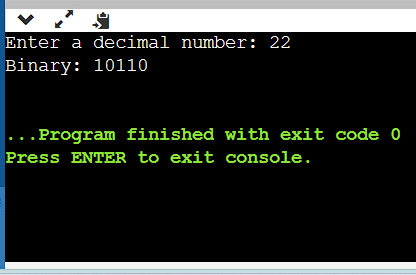
printf("Enter a decimal number: ");

scanf("%d", &decimal);

decimalToBinary(decimal);

return 0;

}



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

#include <stdio.h>

int main() {

int decimal, octal[100], i = 0;

printf("Enter a decimal number: ");

scanf("%d", &decimal);

while (decimal > 0) {

octal[i] = decimal % 8;

decimal /= 8;

i++;

}

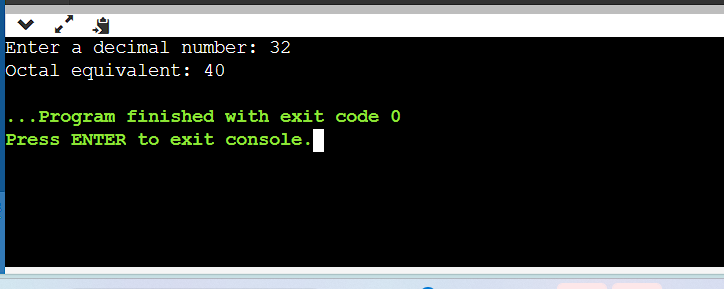
printf("Octal equivalent: ");

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

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

}

return 0;

}

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

#include <stdio.h>

int main() {

int decimal, remainder, i = 0;

char hexadecimal[100];

printf("Enter a decimal number: ");

scanf("%d", &decimal);

while (decimal > 0) {

remainder = decimal % 16;

if (remainder < 10) {

hexadecimal[i] = remainder + '0';

} else {

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

}

decimal /= 16;

i++;

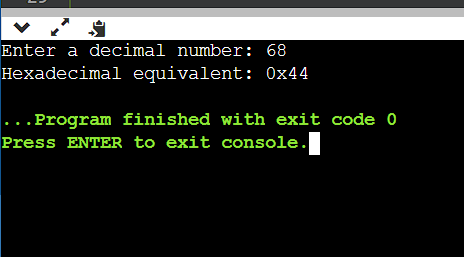
}

printf("Hexadecimal equivalent: 0x");

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

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

}

 return 0;

}

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

#include <stdio.h>

#include <string.h>

int main() {

char hexadecimal[100];

printf("Enter a hexadecimal number: ");

scanf("%s", hexadecimal);

int length = strlen(hexadecimal);

printf("Binary equivalent: ");

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

char hexDigit = hexadecimal[i];

int decimalValue;

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

decimalValue = hexDigit - '0';

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

decimalValue = hexDigit - 'A' + 10;

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

decimalValue = hexDigit - 'a' + 10;

} else {

printf("Invalid hexadecimal input.\n");

return 1;

}

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

if (decimalValue & (1 << j)) {

printf("1");

} else {

printf("0");

}

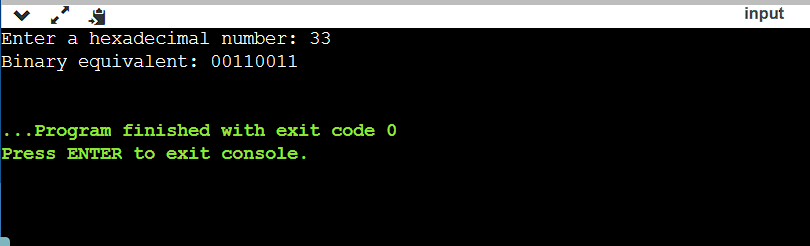
}

}

printf("\n");

return 0;

}



71. Write a C program to convert Hexadecimal

to Octal number system

#include <stdio.h>

#include <string.h>

int main() {

char hex[20];

printf("Enter a hexadecimal number: ");

scanf("%s", hex);

long int dec = 0, oct = 0, i = 0;

// Convert hexadecimal to decimal

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

int digit;

if (hex[j] >= '0' && hex[j] <= '9')

digit = hex[j] - '0';

else if (hex[j] >= 'A' && hex[j] <= 'F')

digit = hex[j] - 'A' + 10;

else if (hex[j] >= 'a' && hex[j] <= 'f')

digit = hex[j] - 'a' + 10;

dec += digit \* (1 << (4 \* i));

i++;

}

i = 1;

// Convert decimal to octal

while (dec != 0) {

oct += (dec % 8) \* i;

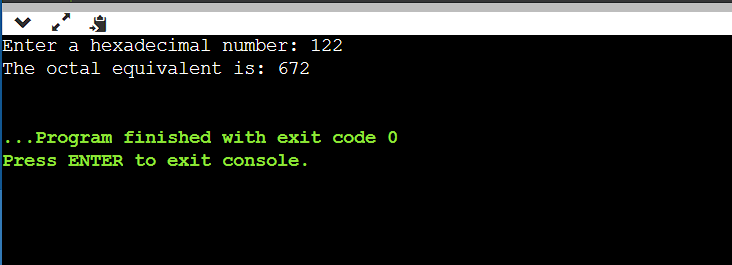
dec /= 8;

i \*= 10;

}

printf("The octal equivalent is: %lo\n", oct);

return 0;

}

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

#include <stdio.h>

#include <math.h>

int main() {

char hexNum[10];

int decimalNum = 0;

int i, j, len;

printf("Enter a hexadecimal number: ");

scanf("%s", hexNum);

// Find the length of the hexadecimal number

len = strlen(hexNum);

// Iterate through the hexadecimal number in reverse order

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

len--;

if (hexNum[i] >= '0' && hexNum[i] <= '9')

j = hexNum[i] - '0';

else if (hexNum[i] >= 'a' && hexNum[i] <= 'f')

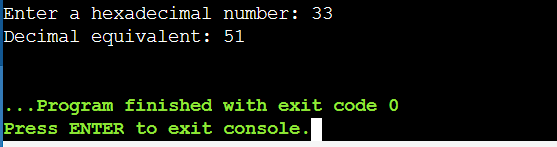
j = hexNum[i] - 'a' + 10;

else if (hexNum[i] >= 'A' && hexNum[i] <= 'F')

j = hexNum[i] - 'A' + 10;

decimalNum += j \* pow(16, len);

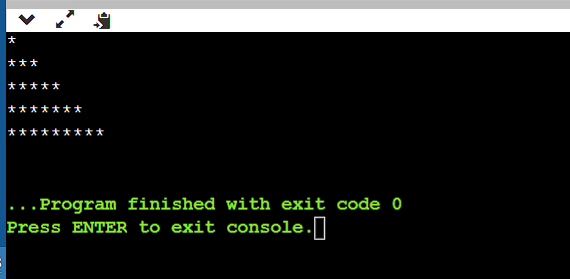
}

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

return 0;

}

Pattern Exercises

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

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

Pyramid Star Pattern

#include <stdio.h>

void printStarPattern(int n) {

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

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

printf("\*");

}

printf("\n");

}

}

int main() {

int n = 5; // Change this value to adjust the number of rows

printStarPattern(n);

return

}

2. \*

\* \*

\* \*

\* \*

\*\*\*\*\*\*\*\*\*

Hollow Pyramid Star Pattern

#include <stdio.h>

void printHollowPyramid(int n) {

int i, j;

// Print upper part of the pyramid

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

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

printf(" ");

}

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

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

printf("\*");

}

else {

printf(" ");

}

}

printf("\n");

}

}

int main() {

int n;

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

scanf("%d", &n);

if(n <= 0) {

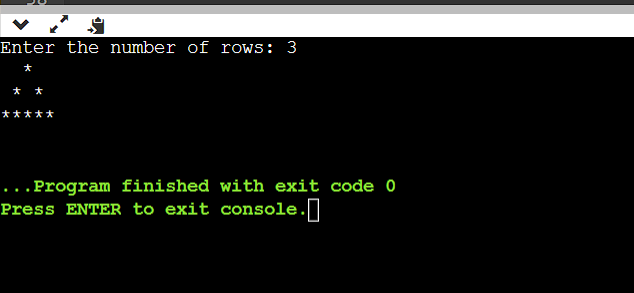
printf("Invalid input. Please enter a positive integer.\n");

return 1;

}

printHollowPyramid(n);

return 0;

}

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

3.Inverted Pyramid Star Pattern

#include <stdio.h>

int main() {

int i, j, rows;

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

scanf("%d", &rows);

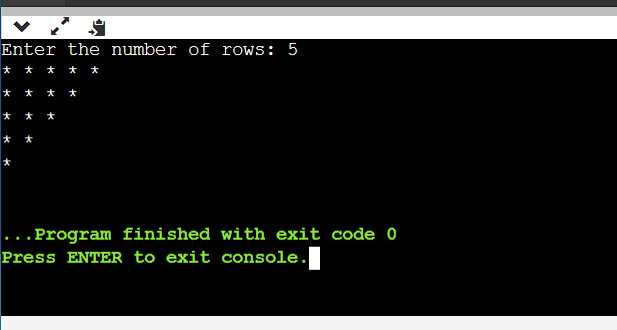
for(i = rows; i >= 1; --i) {

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

printf("\* ");

}

printf("\n");

 }

return 0;

}

\*\*\*\*\*\*\*\*\*

\*\*\*\*

\* \*

\* \*

\* \*

\*

4.Hollow Inverted Pyramid Star Pattern

#include <stdio.h>

int main() {

int rows, i, j;

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

scanf("%d", &rows);

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

// Print spaces

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

printf(" ");

}

// Print stars for the first row or the last row

if(i == 1 || i == rows) {

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

printf("\*");

}

} else { // Print stars for other rows

printf("\*");

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

printf(" ");

}

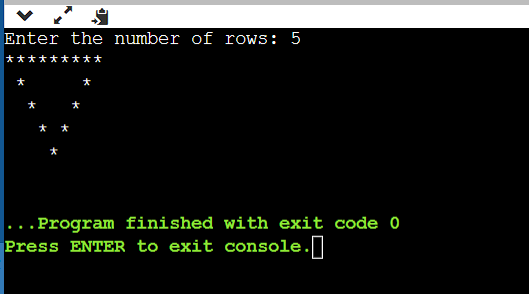
printf("\*");

}

printf("\n");

}

return 0;

}

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

5.Half Diamond Star Pattern

#include <stdio.h>

int main() {

int i, j, rows;

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

scanf("%d", &rows);

// Print upper half of the diamond

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

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

printf("\*");

}

printf("\n");

}

// Print lower half of the diamond

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

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

printf("\*");

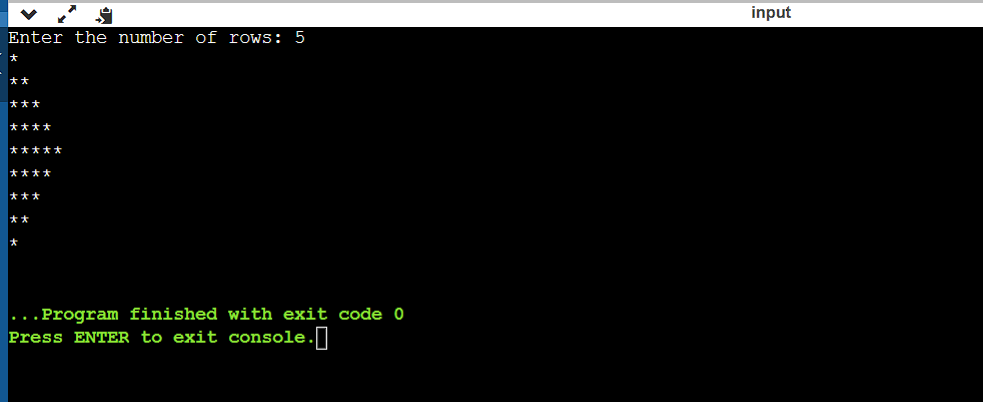
}

printf("\n");

}

return 0;

}



\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

6.Mirrored Half Diamond Star Pattern

#include <stdio.h>

void printHalfDiamond(int n) {

int i, j;

// Upper half of the pattern

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

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

printf("\*");

}

printf("\n");

}

// Lower half of the pattern

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

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

printf("\*");

}

printf("\n");

}

}

int main() {

int n;

// Get user input for the number of rows

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

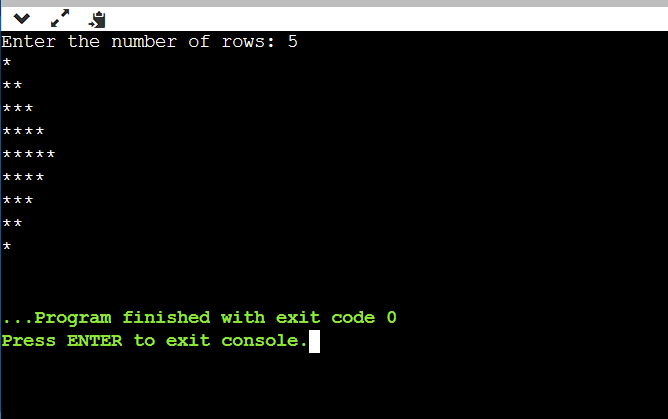
scanf("%d", &n);

// Call the function to print the pattern

printHalfDiamond(n);

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() {

int i, j;

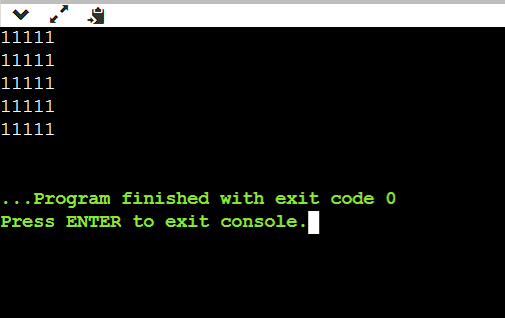
for(i = 0; i < 5; i++) {

for(j = 0; j < 5; j++) {

printf("1");

}

printf("\n");

 }

return 0;

}

Number pattern 1

11111

00000

11111

00000

11111

#include <stdio.h>

int main() {

int i, j;

for (i = 1; i <= 5; i++) {

for (j = 1; j <= 5; j++) {

if (i % 2 == 1) {

printf("1");

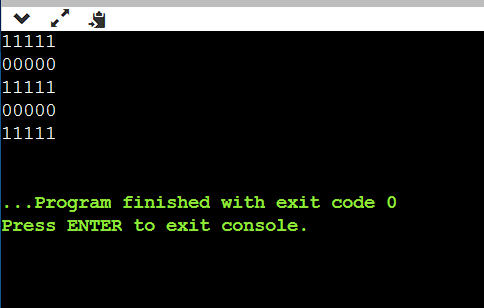
} else {

printf("0");

}

}

printf("\n");

 }

return 0;

}

Number pattern 2

01010

01010

01010

01010

01010

#include <stdio.h>

int main() {

int i, j;

for(i = 1; i <= 5; i++) {

for(j = 1; j <= 5; j++) {

if(j % 2 == 0) {

printf("1");

} else {

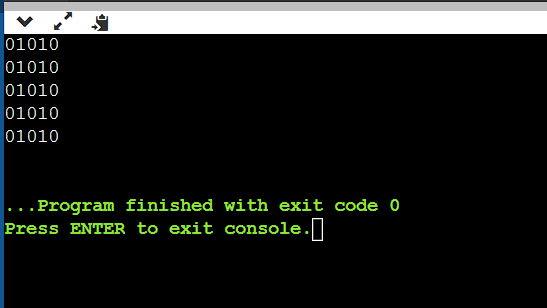
printf("0");

}

}

printf("\n");

}

 return 0;

}

Number pattern 3

11111

10001

10001

10001

11111

#include <stdio.h>

int main() {

int n = 5; // Number of rows and columns

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

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

if (i == 0 || i == n - 1 || j == 0 || j == n - 1)

printf("1");

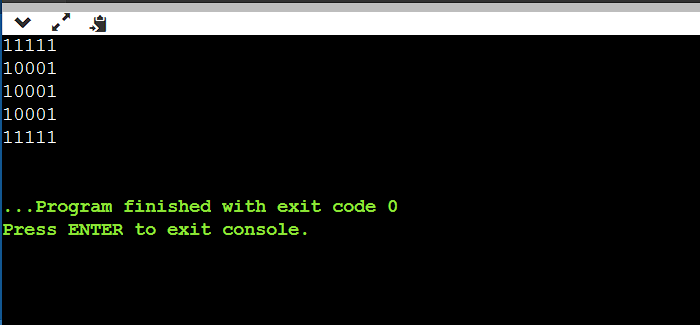
else

printf("0");

}

printf("\n");

}



return 0;

}

Number pattern 4

11111

11111

11011

11111

11111

#include <stdio.h>

int main() {

int n = 5; // Size of the pattern (5x5 in this case)

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

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

if (i == n/2 && j == n/2) { // If we're at the center, print 0

printf("0");

} else {

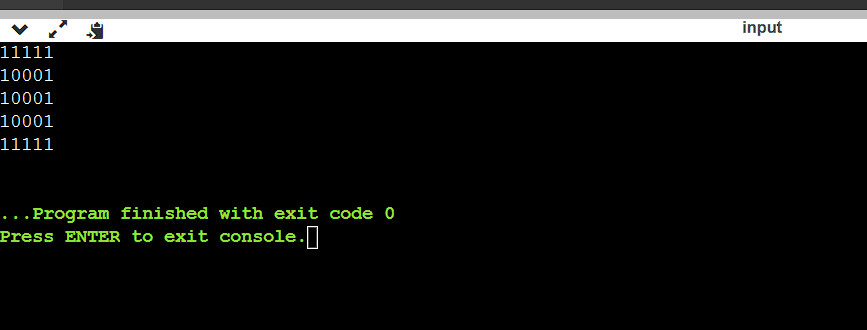
printf("1");

}

}

printf("\n"); // Move to the next line after each row

}

 return 0;

}

Number pattern 5

10101

01010

10101

01010

10101

#include <stdio.h>

int main() {

int rows = 5;

int cols = 5;

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

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

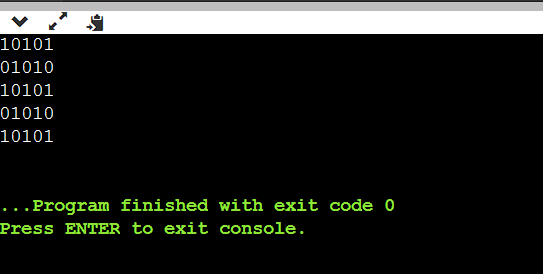
if((i+j) % 2 == 0) {

printf("1");

} else {

printf("0");

}

 }

printf("\n");

}

return 0;

}

**If…Else Exercises**

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

#include <stdio.h>

int main() {

int num1, num2;

// Input the two numbers

printf("Enter first number: ");

scanf("%d", &num1);

printf("Enter second number: ");

scanf("%d", &num2);

// Compare the numbers

if (num1 > num2) {

printf("%d is the maximum\n", num1);

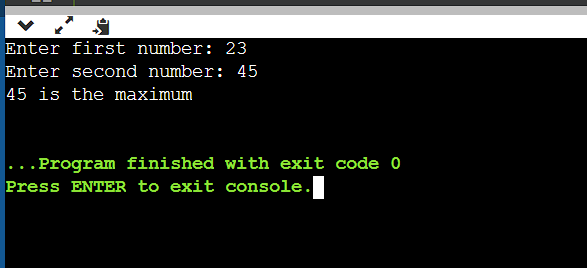
} else {

printf("%d is the maximum\n", num2);

}

return 0;

}



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

#include <stdio.h>

int findMax(int a, int b, int c) {

int max = a;

if (b > max) {

max = b;

}

if (c > max) {

max = c;

}

return max;

}

int main() {

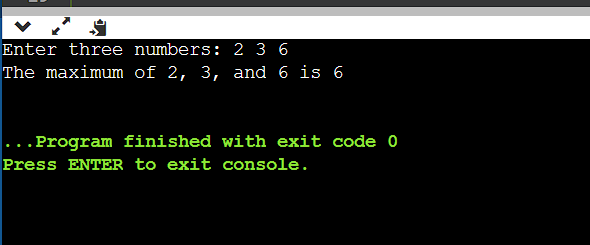
int num1, num2, num3;

printf("Enter three numbers: ");

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

int max = findMax(num1, num2, num3);

printf("The maximum of %d, %d, and %d is %d\n", num1, num2, num3, max);

 return 0;

}

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

#include <stdio.h>

int main() {

int num;

// Read the number from the user

printf("Enter a number: ");

scanf("%d", &num);

if (num > 0) {

printf("%d is positive.\n", num);

} else if (num < 0) {

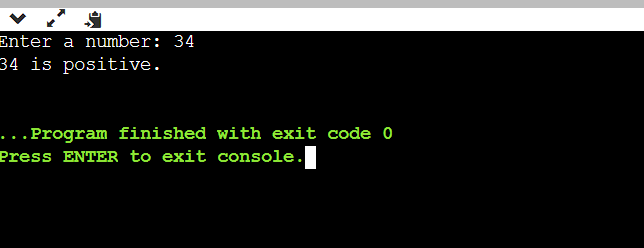
printf("%d is negative.\n", num);

} else {

printf("%d is zero.\n", num);

}

return 0;

}

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;

// Read input from user

printf("Enter a number: ");

scanf("%d", &num);

// Check if the number is divisible by both 5 and 11

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

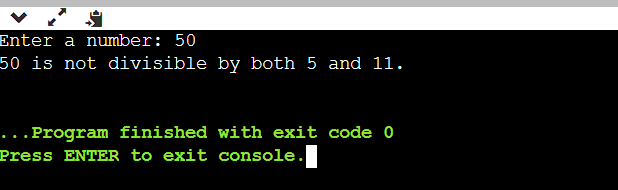
printf("%d is divisible by both 5 and 11.\n", num);

} else {

printf("%d is not divisible by both 5 and 11.\n", num);

}

return 0;

}

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

#include <stdio.h>

int main() {

int num;

// Prompt the user to enter a number

printf("Enter an integer: ");

scanf("%d", &num);

// Check if the number is even or odd

if (num % 2 == 0) {

printf("%d is even.\n", num);

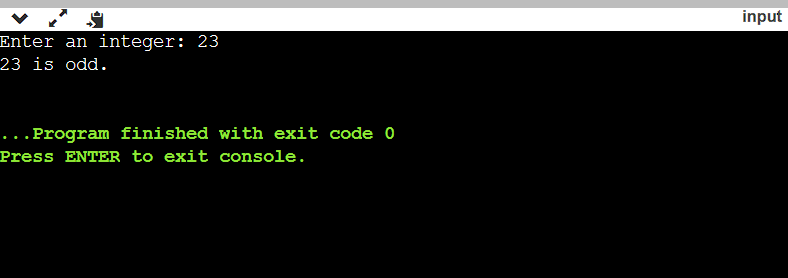
} else {

printf("%d is odd.\n", num);

}

return 0;

}



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

#include <stdio.h>

int main() {

int year;

// Input year from user

printf("Enter a year: ");

scanf("%d", &year);

// Check if the year is a leap year

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

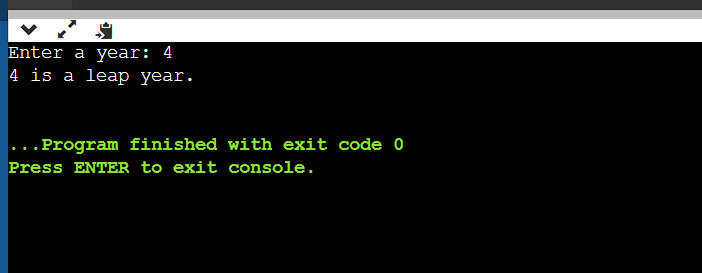
printf("%d is a leap year.\n", year);

} else {

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

}

return 0;

}

7. Write a C program to check whether a character is alphabet or not #include <stdio.h>

int main() {

char ch;

// Input character from user

printf("Enter a character: ");

scanf("%c", &ch);

// Check if the character is an alphabet

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

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

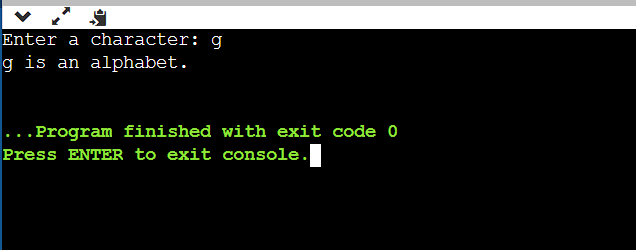
} else {

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

}

return 0;

}



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

#include <stdio.h>

int main() {

char ch;

printf("Enter an alphabet: ");

scanf("%c", &ch);

// Using switch case to check if it's a vowel or consonant

switch(ch) {

case 'a':

case 'e':

case 'i':

case 'o':

case 'u':

case 'A':

case 'E':

case 'I':

case 'O':

case 'U':

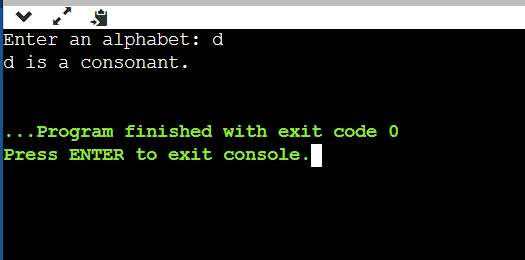
printf("%c is a vowel.\n", ch);

break;

default:

printf("%c is a consonant.\n", ch);

}



return 0;

}

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

character.

#include <stdio.h>

int main() {

char character;

// Ask user to enter a character

printf("Enter a character: ");

scanf("%c", &character);

// Check if the character is an alphabet

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

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

}

// Check if the character is a digit

else if (character >= '0' && character <= '9') {

printf("%c is a digit.\n", character);

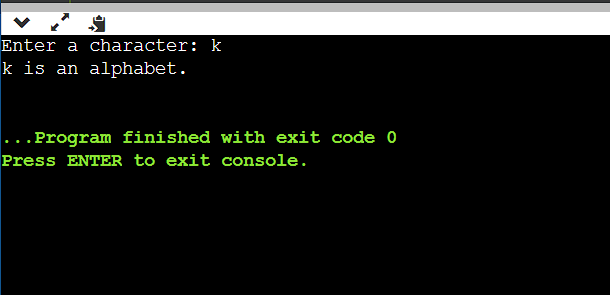
}

// If it's not an alphabet or a digit, it's a special character

else {

printf("%c is a special character.\n", character);

}

 return 0;

}