LAB FILE

INTRODUCTION TO C PROGRAMMING



BATCH : 2023 – 2027

BCA (HONS.) WITH AI AND DS

Submitted By: Krish Grover Submitted to: Ms. Pooja Chahar

Student Id: 23151173

INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S. NO | PROGAM | PAGENO | SIGN | REMARKS |
| 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 numbers by using third variable. |  |  |  |
| 8 | WAP to SWAP two variables numbers without using third variable. |  |  |  |
| 9 | WAP to SWAP three variables numbers without using third variable. |  |  |  |
| 10 | WAP to find the area of rectangle. |  |  |  |
| 11 | WAP to find the 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 operators. |  |  |  |
| 22 | WAP to identify gender in single character and print full gender. |  |  |  |
| 23 | Write a C program to print all natural numbers in reverse. |  |  |  |
| 24 | Write a C program to print all alphabet from a to z. |  |  |  |
| 25 | Write a C program to print all natural numbers. |  |  |  |
| 26 | Write a C program to print all even numbers between 1 to 100. |  |  |  |
| 27 | Write a C program to print all odd numbers 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. |  |  |  |
| 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 |  |  |  |
| 51 | Write a C program to find all prime factors of a number. |  |  |  |
| 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 convert Hexadecimal to Decimal number system. |  |  |  |

Pattern Exercises

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Star pattern programs - Write a C program to print the given star patterns   1. Pyramid Star Pattern 2. Hollow Pyramid Star Pattern 3. Inverted Pyramid Star Pattern 4. Hollow Inverted Pyramid Star Pattern 5. Half Diamond Star Pattern 6. Mirrored Half Diamond Star Pattern |  |  |  |
| 2 | Number pattern programs - Write a C program to print the given number patterns   1. Square Number Patterns 2. Number Pattern 1 3. Number Pattern 2 4. Number Pattern 3 5. Number Pattern 4 6. Number Pattern 5 |  |  |  |

If-Else Exercise

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Write a C program to find maximum between two numbers. |  |  |  |
| 2 | Write a C program to find maximum between three numbers. |  |  |  |
| 3 | Write a C program to check whether a number is negative, positive or zero. |  |  |  |
| 4 | Write a C program to check whether a number is divisible by 5 and 11 or not. |  |  |  |
| 5 | Write a C program to check whether a number is even or odd. |  |  |  |
| 6 | Write a C program to check whether a year is leap year or not. |  |  |  |
| 7 | Write a C program to check whether a character is alphabet or not. |  |  |  |
| 8 | Write a C program to input any alphabet and check whether it is vowel or consonant. |  |  |  |
| 9 | Write a C program to input any character and check whether it is alphabet, digit or special  Character. |  |  |  |
| 10 | Write a C program to check whether a character is uppercase or lowercase alphabet. |  |  |  |
| 11 | Write a C program to input week number and print week day |  |  |  |
| 12 | Write a C program to input month number and print number of days in that month. |  |  |  |
| 13 | Write a C program to count total number of notes in given amount. |  |  |  |
| 14 | Write a C program to input month number and print number of days in that month. |  |  |  |
| 15 | Write a C program to count total number of notes in given amount. |  |  |  |
| 16 | Write a C program to input angles of a triangle and check whether triangle is valid or not. |  |  |  |
| 17 | Write a C program to input all sides of a triangle and check whether triangle is valid or not. |  |  |  |
| 18 | Write a C program to check whether the triangle is equilateral, isosceles or scalene triangle. |  |  |  |
| 19 | Write a C program to find all roots of a quadratic equation. |  |  |  |
| 20 | Write a C program to calculate profit or loss. |  |  |  |
| 21 | Write a C program to input marks of five subjects Physics, Chemistry, Biology, Mathematics  and Computer. Calculate percentage and grade according to the 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 |  |  |  |
| 22 | 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% |  |  |  |
| 23 | 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. |  |  |  |
| 24 | Write a C program to convert specified days into years, weeks and days. |  |  |  |

Array Exercises

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1**.** | Write a program in C to read n number of values in an array and display them in reverse order. |  |  |  |
| 2. | Write a program in C to find the sum of all elements of the array. |  |  |  |
| 3. | Write a program in C to copy the elements of one array into another array |  |  |  |
| 4. | Write a program in C to count the total number of duplicate elements in an array |  |  |  |
| 5. | Write a program in C to find the maximum and minimum elements in an array |  |  |  |
| 6. | Write a C program to sort the elements of an array in descending order. |  |  |  |
| 7. | Write a program in C to separate odd and even integers into separate arrays |  |  |  |
| 8. | Write a program in C to merge two arrays of the same size sorted in descending/ascending order. |  |  |  |
| 9**.** | Write a program in C to merge two arrays of the same size sorted in descending order. |  |  |  |
| 10**.** | WAP using Switch case: Consider two matrices of the size m and n. Implement matrix operation and display. Show these things in program 1) Read matrix elements and display 2) Matrix Multiplication and display 3) addition of matrix and display 4)Subtraction of Matrix and display 5)Transpose of Matrix and display |  |  |  |

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

Program.

#include <stdio.h>

int main()

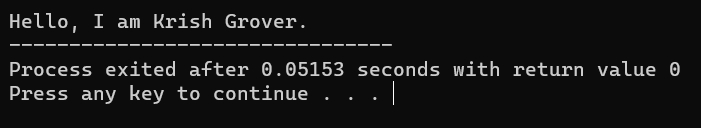
{

// printf() displays the content inside

printf("Hello, I am Krish Grover.");

return 0;

}



1. WAP to add two numbers.

#include <stdio.h>

int main()

{

int number1, number2, sum;

printf("Enter two integers: ");

scanf("%d %d", &number1, &number2);

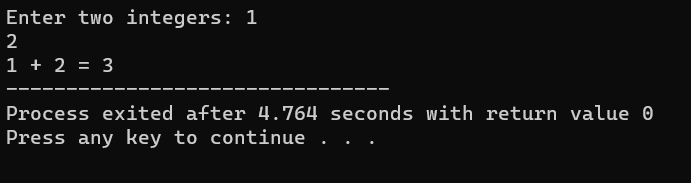
// calculate the sum

sum = number1 + number2;

printf("%d + %d = %d", number1, number2, sum);

return 0;

}



1. WAP to find area of circle.

#include <stdio.h>

#include <math.h>

int main()

{

float radius, area;

printf("Enter the radius of a circle\n");

scanf("%f", &radius);

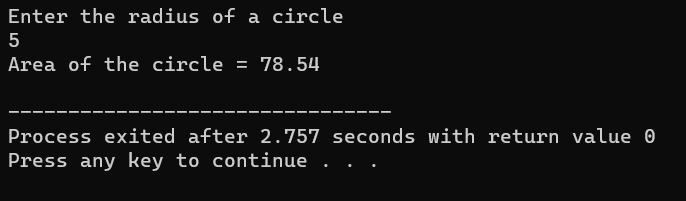
area = 3.14159\*radius\*radius;

printf("Area of the circle = %.2f\n", area);

// printing upto two decimal places

return 0;

}



1. WAP to divide two numbers.

#include<stdio.h>

int main()

{

int a, b, c;

printf("Enter 2 numbers for division\n");

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

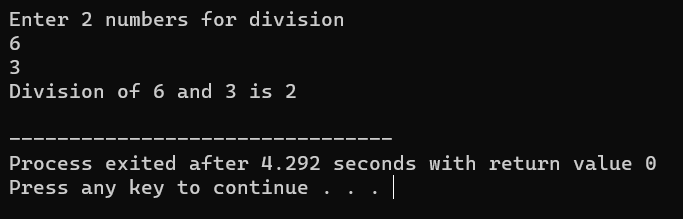
c = a / b;

printf("Division of %d and %d is %d\n", a, b,

c);

return 0;

}



1. WAP to print ASCII value.

#include <stdio.h>

int main()

{

char c;

printf("Enter a character: ");

scanf("%c", &c);

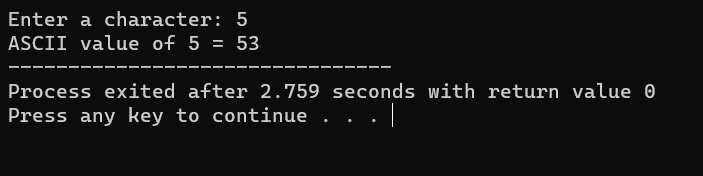
// %d displays the integer value of character

// %c displays the actual character

printf("ASCII value of %c = %d", c, c);

return 0;

}



1. WAP to multiply floating point numbers.

#include <stdio.h>

int main() {

double a, b, product;

printf("Enter two numbers: ");

scanf("%lf %lf", &a, &b);

// Calculating product

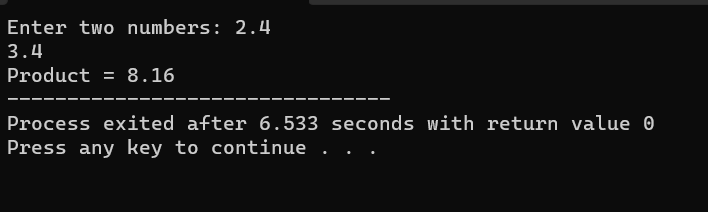
product = a \* b;

// %.2lf displays number up to 2 decimal point

printf("Product = %.2lf", product);

return 0;

}



1. WAP to SWAP two variables number by using

third variable .

#include<stdio.h>

int main() {

double first, second, temp;

printf("Enter first number: ");

scanf("%lf", &first);

printf("Enter second number: ");

scanf("%lf", &second);

// value of first is assigned to temp

temp = first;

// value of second is assigned to first

first = second;

// value of temp (initial value of first) is assigned to second

second = temp;

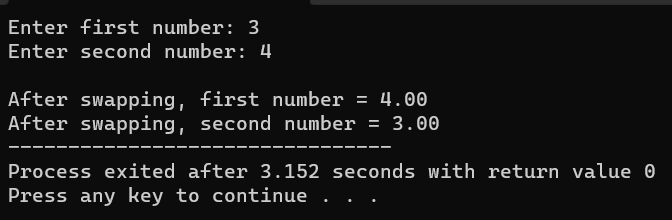
// %.2lf displays number up to 2 decimal points

printf("\nAfter swapping, first number = %.2lf\n", first);

printf("After swapping, second number = %.2lf", second);

return 0;

}



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

#include<stdio.h>

int main()

{

int x, y;

printf("Program by Krish Grover");

// Prompt user to input values for x and y

printf("\nInput value for x & y: \n");

scanf("%d%d",&x,&y);

// Display the values of x and y before swapping

printf("Before swapping the value of x & y: %d %d",x,y);

// Swap the values of x and y using arithmetic operations

x = x + y;

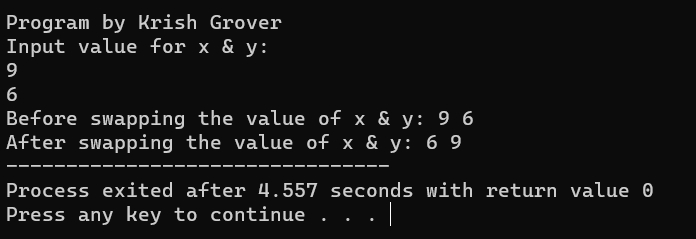
y = x - y;

x = x - y;

// Display the values of x and y after swapping

printf("\nAfter swapping the value of x & y: %d %d",x,y);

return 0;



1. WAP to SWAP three vairable numbers without using third variables.

#include <stdio.h>

// Assign c's value to a, a's value to b and

// b's value to c.

int main()

{

int a, b ,c;

printf("Program by Krish Grover");

printf("\nEnter Three Numbers : ");

scanf("\n%d%d%d",&a,&b,&c);

printf( " \nBefore swapping a = %d , b = %d, c = %d ",a,b,c);

// Store sum of all in a

a = a + b + c; // (a = 60)

// After this, b has value of a

b = a - (b+c); // (b = 60 – (20+30) =10)

// After this, c has value of b

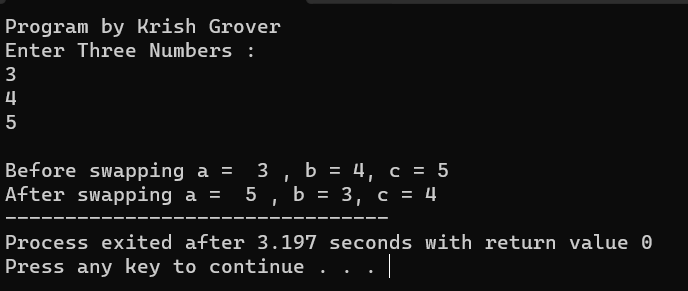
c = a - (b+c); // (c = 60 – (10 + 30) = 20)

// After this, a has value of c

a = a - (b+c); //(a = 60 – (10 + 20) = 30)

printf( "\nAfter swapping a = %d , b = %d, c = %d ",a,b,c);

return 0;}



1. Wap to find the area of rectangle.

#include <stdio.h>

int main()

{

int width;

int length;

printf("Program by Krish Grover");

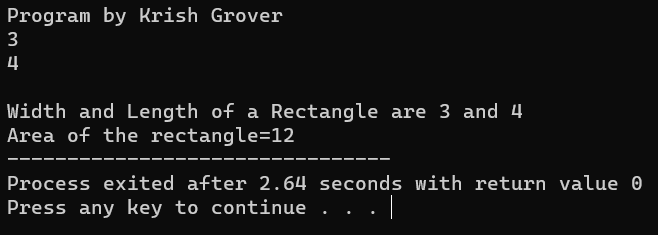
scanf("\n%d%d",&width,&length);

printf("\nWidth and Length of a Rectangle are %d and %d",width,length);

int area=width\*length;

printf("\nArea of the rectangle=%d",area);

}



1. WAP to find area of square.

#include<stdio.h>

int main()

{

int side, area;

printf("Program by Krish Grover");

printf("\nEnter the Length of Side : ");

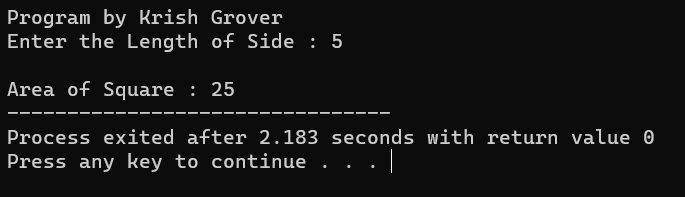
scanf("%d", &side);

area = side \* side;

printf("\nArea of Square : %d", area);

return (0);

}



1. WAP to find area of right angle triangle, isosceles triangle, any triangle with 3 sides.

#include <stdio.h>

#include <math.h>

int main()

{

double a, b, c, s, area;

printf("Program by Krish Grover");

printf("\nEnter sides of a triangle\n");

scanf("%lf%lf%lf", &a, &b, &c);

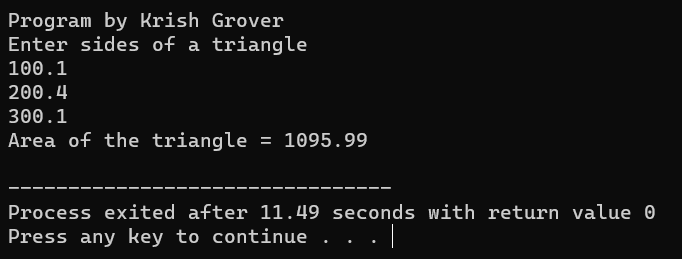
s = (a+b+c)/2; // Semiperimeter

area = sqrt(s\*(s-a)\*(s-b)\*(s-c));

printf("Area of the triangle = %.2lf\n", area);

return 0;

}



1. WAP to find Area and Volume of Cube.

#include <stdio.h>

#include <math.h>

int main()

{

float side, surfacearea, volume;

printf("Program by Krish Grover");

printf("\nEnter the length of a side \n");

scanf("%f", &side);

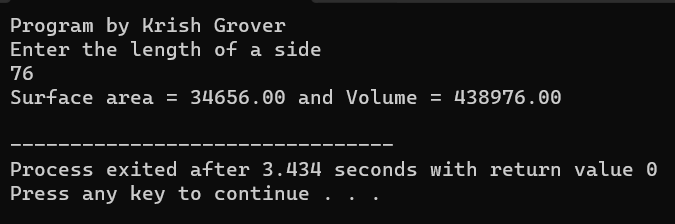
surfacearea = 6.0 \* side \* side;

volume = pow(side, 3);

printf("Surface area = %6.2f and Volume = %6.2f \n", surfacearea,

volume);

}



1. WAP to find area and volume of cuboid.

#include <stdio.h>

#include <math.h>

int main()

{

float width, length, height;

float surfacearea, volume;

printf("Program by Krish Grover");

printf("\nEnter value of width, length & height of the cuboids:\n");

scanf("%f%f%f", &width, &length, &height);

surfacearea = 2 \*(width \* length + length \* height +

height \* width);

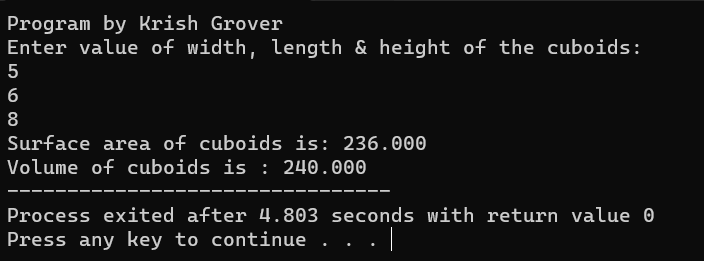
volume = width \* length \* height;

printf("Surface area of cuboids is: %.3f", surfacearea);

printf("\nVolume of cuboids is : %.3f", volume);

return 0;

}



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

#include <stdio.h>

int main()

{

double n1, n2, n3;

printf("Program by Krish Grover");

printf("\nEnter three different numbers: ");

scanf("%lf %lf %lf", &n1, &n2, &n3);

// if n1 is greater than both n2 and n3, n1 is the largest

if (n1 >= n2 && n1 >= n3)

printf("%.2f is the largest number.", n1);

// if n2 is greater than both n1 and n3, n2 is the largest

if (n2 >= n1 && n2 >= n3)

printf("%.2f is the largest number.", n2);

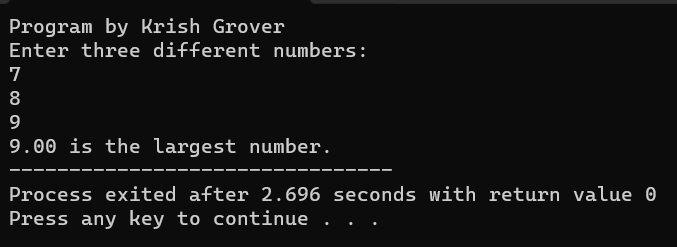
// if n3 is greater than both n1 and n2, n3 is the largest

if (n3 >= n1 && n3 >= n2)

printf("%.2f is the largest number.", n3);

return 0;

}



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

#include <stdio.h>

#include <string.h>

int main()

{

int flag1 = 0,flag2 = 0;

char user\_name[] = "Krish Grover";

char user\_password[] = "kg@123";

char user\_name1[20];

char user\_password1[20];

printf("Program by Krish Grover");

printf("\nEnter Username :--> ");

gets(user\_name1);

printf("Enter Password :--> ");

gets(user\_password1);

if((strcmp(user\_name, user\_name1) == 0) && (strcmp(user\_password, user\_password1) ==0 ))

{

printf("Success");

}

else

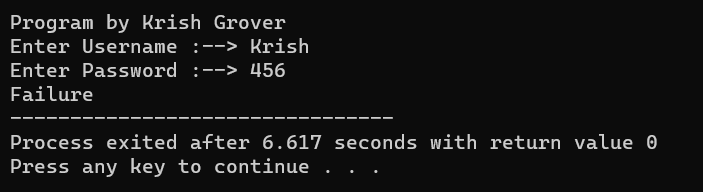
{

printf("Failure");

}

return 0;

}



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

#include <stdio.h>

int main ()

{

// declare local variable

int num, bit;

printf("Program by Krish Grover");

printf ("\nEnter a positive number: ");

scanf (" %d", &num);

printf ("No. of binary bits shifted to the left side: ");

scanf (" %d", &bit);

// use left shift operator to shift the bits

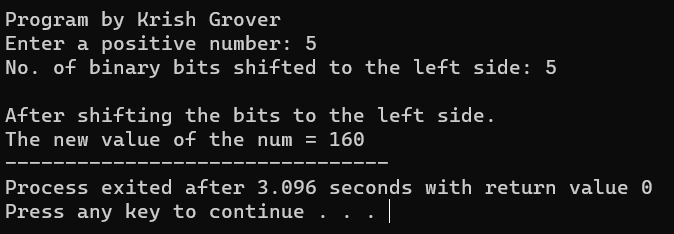
num = (num << bit);

printf (" \nAfter shifting the bits to the left side. ");

printf (" \nThe new value of the num = %d", num);

return 0;

}



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

#include <stdio.h>

int main ()

{

// declare local variable

int num, bit;

printf("Program by Krish Grover");

printf ("\nEnter a positive number: ");

scanf (" %d", &num);

printf ("No. of binary bits shifted to the right side: ");

scanf ("%d", &bit);

// use right shift operator to shift the bits

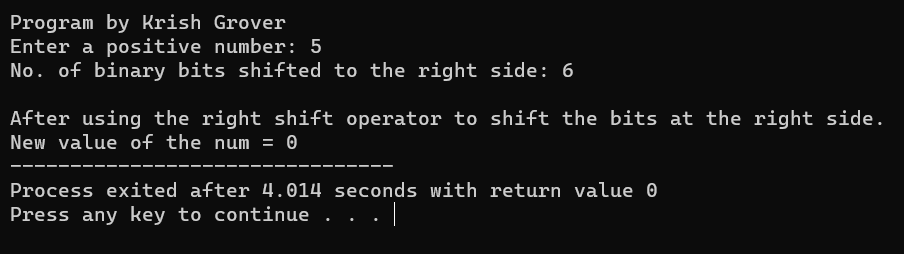
num = (num >> bit);

printf (" \nAfter using the right shift operator to shift the bits at the right side. ");

printf (" \nNew value of the num = %d", num);

return 0;

}



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

#include <stdio.h>

int main()

{

int num1, num2;

printf("Program by Krish Grover");

printf("\nEnter the first and second integer: ");

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

printf("Pre-increment:\n");

printf("Original value of num1 and num2: %d and %d\n", num1, num2);

printf("Updated value of num1 after pre-increment: %d, %d\n", ++num1, ++num2);

num1 = num1 - 1;

num2 = num2 - 1;

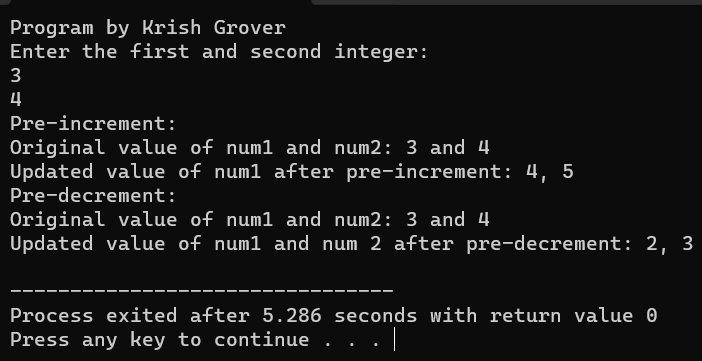
printf("Pre-decrement:\n");

printf("Original value of num1 and num2: %d and %d\n", num1, num2);

printf("Updated value of num1 and num 2 after pre-decrement: %d, %d\n", --num1, --num2);

return 0;

}



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

#include <stdio.h>

int main()

{

int num1, num2;

printf("Program by Krish Grover:");

printf("\nEnter the first and second integer: ");

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

printf("Post-increment:\n");

printf("Original value of num1 and num2: %d and %d\n", num1, num2);

printf("Updated value of num1 after pre-increment: %d, %d\n",num1++, num2++);

num1--;

num2--;

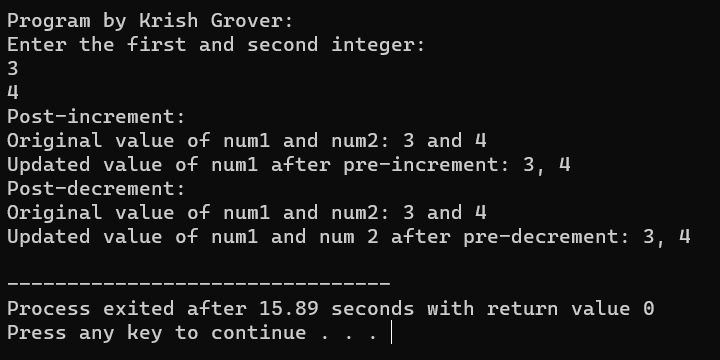
printf("Post-decrement:\n");

printf("Original value of num1 and num2: %d and %d\n", num1, num2);

printf("Updated value of num1 and num 2 after pre-decrement: %d, %d\n",num1--,num2--);

return 0;

}



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

#include <stdio.h>

int divisible(int num)

{

return (num % 7 == 0) || (num % 9 == 0);

}

int main(){

int a;

printf("Program by Krish Grover:");

printf("\nEnter an integer: ");

scanf("%d", &a);

if (divisible(a)){

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

}

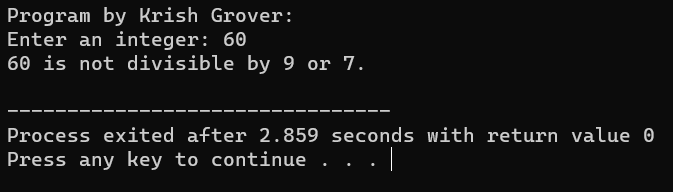
else{

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

}

return 0;

}



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

#include <stdio.h>

void identifyGender(char input){

if (input == 'M' || input == 'm'){

printf("Gender: Male\n");

}

else if (input == 'F' || input == 'f') {

printf("Gender: Female\n");

}

else{

printf("Invalid input. Please enter 'M' or 'F'.\n");

}

}int main(){

char gender;

printf("Program By Krish Grover : ");

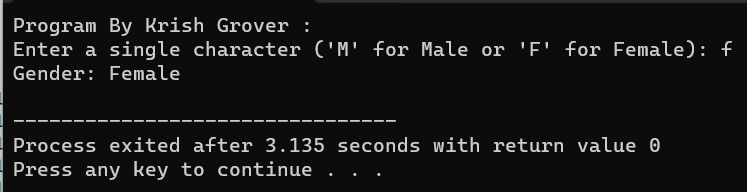
printf("\nEnter a single character ('M' for Male or 'F' for Female): ");

scanf(" %c", &gender);

identifyGender(gender);

return 0;

}



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

#include <stdio.h>

int main() {

int n;

printf("Program By Krish Grover :\n");

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

scanf("%d", &n);

if (n <= 0) {

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

} else {

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

while (n >= 1) {

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

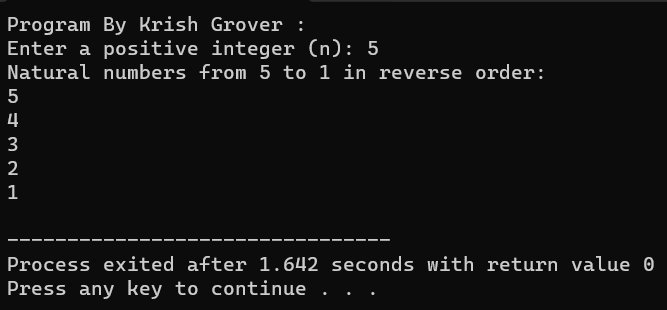
n--;

}

}

return 0;

}



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

#include <stdio.h>

int main() {

char letter;

printf("Program By Krish Grover :\n");

printf("Alphabets from 'a' to 'z':\n");

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

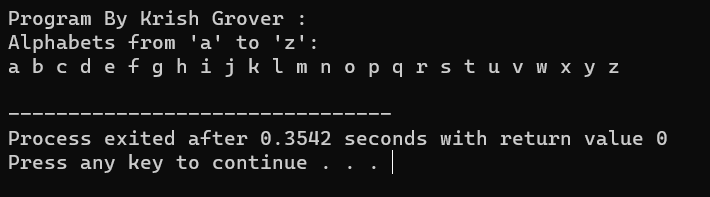
printf("%c ", letter);

}

printf("\n");

return 0;

}



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

#include <stdio.h>

int main() {

int n;

printf("Program By Krish Grover :\n");

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

scanf("%d", &n);

if (n <= 0) {

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

} else {

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

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

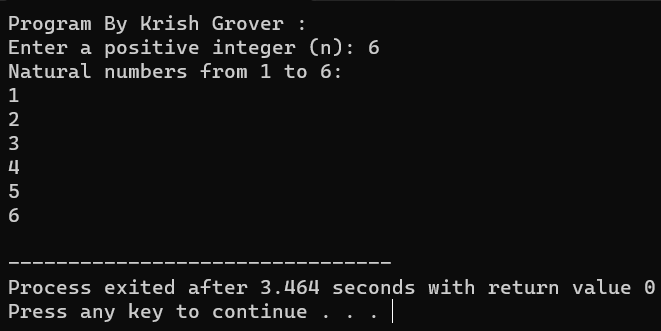
printf("%d\n", i);

}

}

return 0;

}



1. Program to print all even numbers between

1 to 100.

#include <stdio.h>

int main() {

printf("Program By Krish Grover :\n");

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

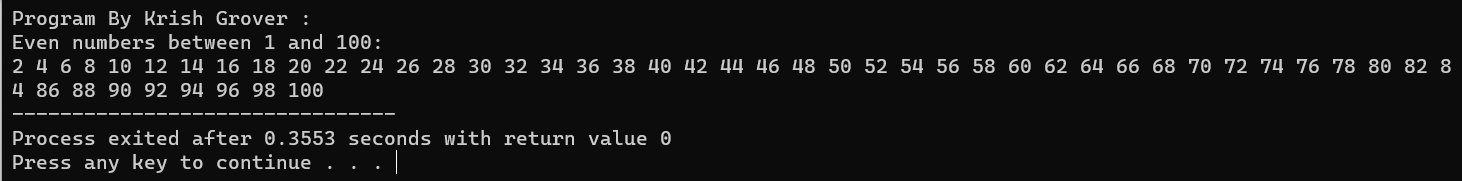
for (int num = 2; num <= 100; num += 2) {

printf("%d ", num);

}

return 0;

}



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

#include <stdio.h>

int main() {

printf("Program By Krish Grover :\n");

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

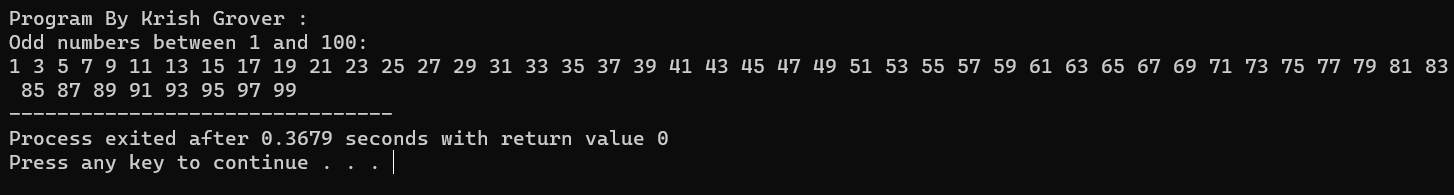
for (int num = 1; num <= 100; num += 2) {

printf("%d ", num);

}

return 0;

}



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

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Program By Krish Grover :\n");

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++) {

sum += i;

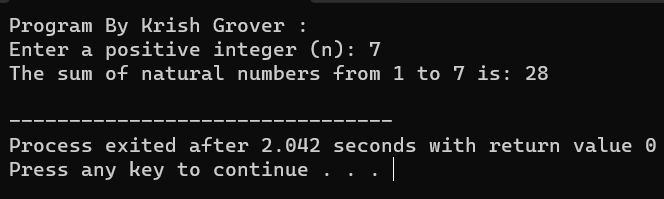
}

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

}

return 0;

}



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

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Program By Krish Grover :\n");

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

scanf("%d", &n);

if (n <= 0) {

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

} else {

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

sum += i;

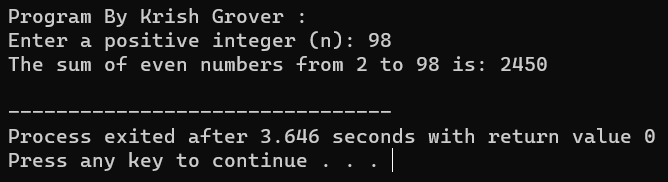
}

printf("The sum of even numbers from 2 to %d is: %d\n", n, sum);

}

return 0;

}



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

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Program By Krish Grover :\n");

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 += 2) {

sum += i;

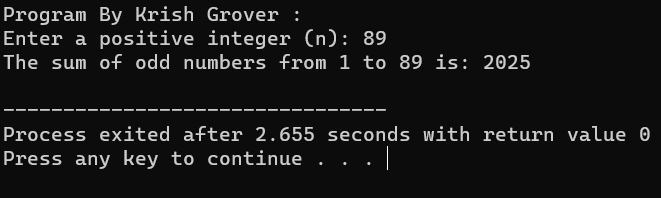
}

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

}

return 0;

}



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

#include <stdio.h>

int main() {

int num, i;

printf("Program By Krish Grover :\n");

printf("Enter a number to print its multiplication table: ");

scanf("%d", &num);

printf("Multiplication table for %d:\n", num);

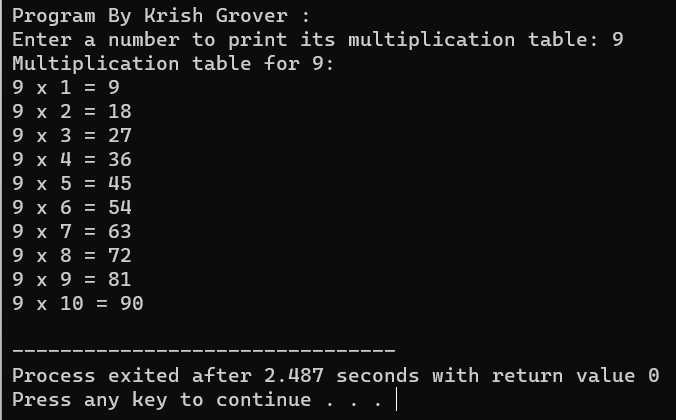
for (i = 1; i <= 10; i++) {

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

}

return 0;

}



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

#include <stdio.h>

int main() {

int num, count = 0;

printf("Program By Krish Grover :\n");

printf("Enter an integer: ");

scanf("%d", &num);

if (num < 0) {

num = -num; // Make the number positive if it's negative

}

if (num == 0) {

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

} else {

while (num > 0) {

count++;

num /= 10;

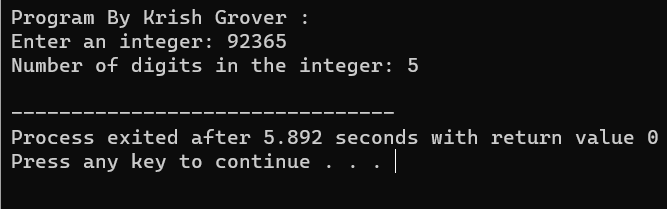
}

}

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

return 0;

}



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

#include <stdio.h>

int main() {

int num, firstDigit, lastDigit;

printf("Program By Krish Grover :\n");

printf("Enter an integer: ");

scanf("%d", &num);

if (num < 0) {

num = -num; // Make the number positive if it's negative

}

// Find the last digit

lastDigit = num % 10;

// Find the first digit

while (num >= 10) {

num /= 10;

}

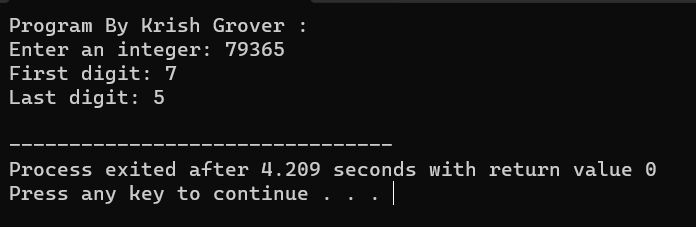
firstDigit = num;

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

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

return 0;

}



1. 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("Program By Krish Grover :\n");

printf("Enter an integer: ");

scanf("%d", &num);

if (num < 0) {

num = -num; // Make the number positive if it's negative

}

// Find the last digit

lastDigit = num % 10;

// Find the first digit

while (num >= 10) {

num /= 10;

}

firstDigit = num;

// Calculate the sum

sum = firstDigit + lastDigit;

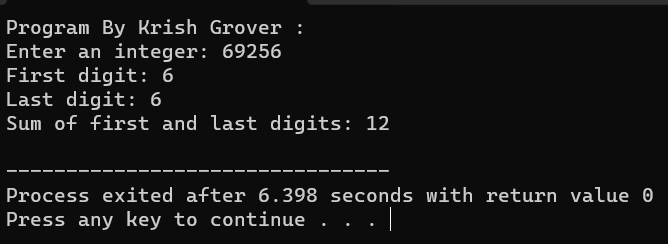
printf("First digit: %d\n", firstDigit);

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

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

return 0;

}



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

#include <stdio.h>

#include <math.h>

int main() {

int num, originalNum, firstDigit, lastDigit, swappedNum = 0;

int numDigits = 0;

printf("Program By Krish Grover :\n");

printf("Enter an integer: ");

scanf("%d", &num);

originalNum = num;

// Count the number of digits in the integer

while (num != 0) {

num /= 10;

numDigits++;

}

num = originalNum; // Reset num to the original value

if (num < 0) {

num = -num; // Make the number positive if it's negative

}

// Find the last digit

lastDigit = num % 10;

// Find the first digit

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

// Swap the first and last digits

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

if (originalNum < 0) {

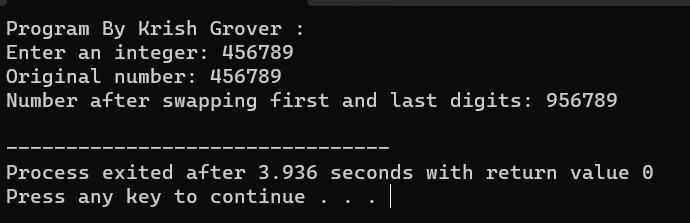
swappedNum = -swappedNum; // Preserve the sign if the original number was negative

}

printf("Original number: %d\n", originalNum);

printf("Number after swapping first and last digits: %d\n", swappedNum);return 0;

}



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

#include <stdio.h>

int main() {

int num, originalNum, digit, sum = 0;

printf("Program By Krish Grover :\n");

printf("Enter an integer: ");

scanf("%d", &num);

originalNum = num;

if (num < 0) {

num = -num; // Make the number positive if it's negative

}

while (num > 0) {

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

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

num /= 10; // Remove the last digit

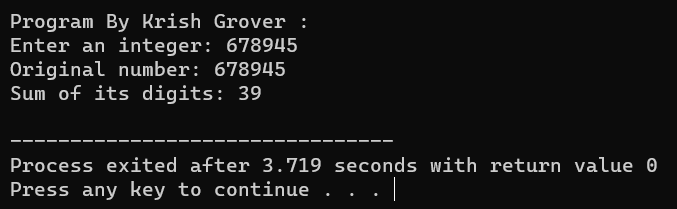
}

printf("Original number: %d\n", originalNum);

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

return 0;

}



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

#include <stdio.h>

int main() {

int num, originalNum, digit, product = 1;

printf("Program By Krish Grover :\n");

printf("Enter an integer: ");

scanf("%d", &num);

originalNum = num;

if (num < 0) {

num = -num; // Make the number positive if it's negative

}

while (num > 0) {

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

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

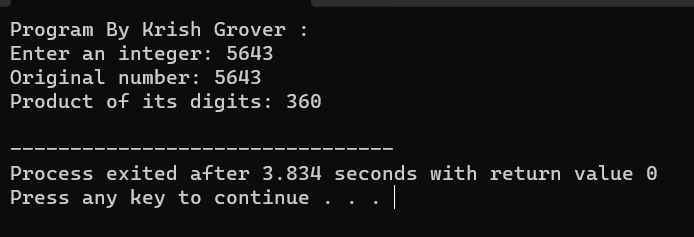
num /= 10; // Remove the last digit

}

printf("Original number: %d\n", originalNum);

printf("Product of its digits: %d\n", product);return 0;

}



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

#include <stdio.h>

int main() {

int num, reversedNum = 0;

printf("Program By Krish Grover :\n");

printf("Enter an integer: ");

scanf("%d", &num);

int originalNum = num;

while (num != 0) {

int digit = num % 10;

reversedNum = reversedNum \* 10 + digit;

num /= 10;

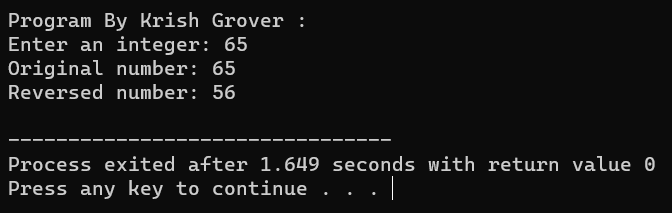
}

printf("Original number: %d\n", originalNum);

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

return 0;

}



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

#include <stdio.h>

int main() {

int num, reversedNum = 0, originalNum;

printf("Program By Krish Grover :\n");

printf("Enter an integer: ");

scanf("%d", &num);

originalNum = num;

while (num != 0) {

int digit = num % 10;

reversedNum = reversedNum \* 10 + digit;

num /= 10;

}

if (originalNum == reversedNum) {

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

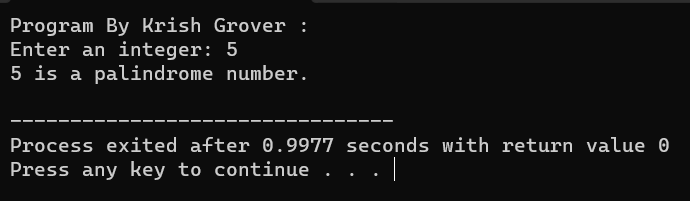
} else {

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

}

return 0;

}



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

#include <stdio.h>

int main() {

int num, digit;

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

printf("Program By Krish Grover :\n");

printf("Enter an integer: ");

scanf("%d", &num);

if (num < 0) {

num = -num; // Make the number positive if it's negative

}

while (num > 0) {

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

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

num /= 10; // Remove the last digit

}

printf("Digit Frequencies:\n");

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

if (digitCount[i] > 0) {

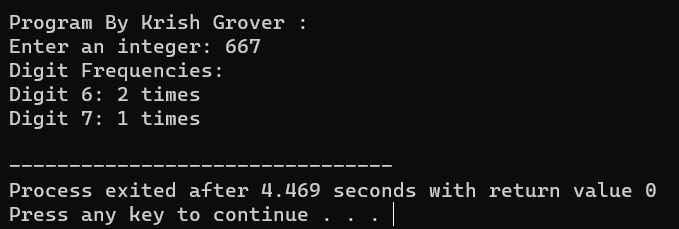
printf("Digit %d: %d times\n", i, digitCount[i]);

}

}

return 0;

}



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

#include <stdio.h>

// Arrays to store the words for numbers 0 to 19 and multiples of 10.

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

"Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};

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

int main() {

int num;

int digit, tens\_digit, ones\_digit;

printf("Program By Krish Grover:\n");

printf("Enter a number: ");

scanf("%d", &num);

if (num < 0 || num > 9999) {

printf("Number out of range (0-9999)\n");

return 1;

}

if (num == 0) {

printf("Zero\n");

return 0;

}

printf("In words: ");

// Extract thousands place

digit = num / 1000;

if (digit > 0) {

printf("%s Thousand ", ones[digit]);

}

// Extract hundreds place

num %= 1000;

digit = num / 100;

if (digit > 0) {

printf("%s Hundred ", ones[digit]);

}

// Extract tens and ones place

num %= 100;

tens\_digit = num / 10;

ones\_digit = num % 10;

if (tens\_digit > 1) {

printf("%s", tens[tens\_digit]);

if (ones\_digit > 0) {

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

}

} else if (tens\_digit == 1) {

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

} else if (ones\_digit > 0) {

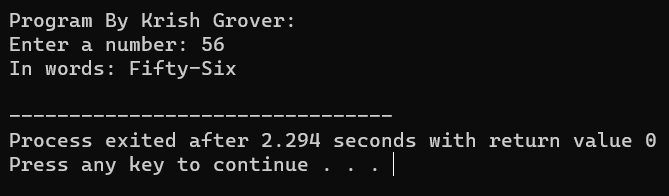
printf("%s", ones[ones\_digit]);

}

printf("\n");

return 0;

}



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

#include <stdio.h>

int main() {

int i;

printf("Program By Krish Grover:\n");

printf("ASCII Character Values:\n");

printf("----------------------------\n");

printf("Decimal Octal Hex Character\n");

printf("----------------------------\n");

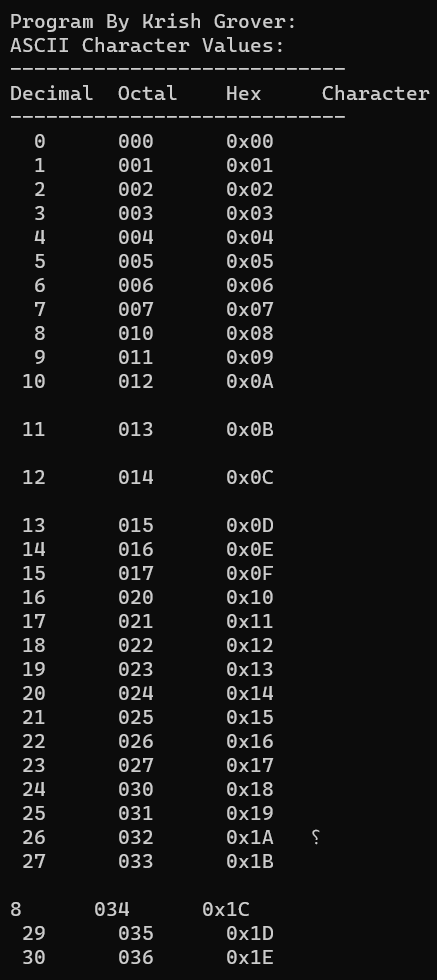
for (i = 0; i <= 127; i++) {

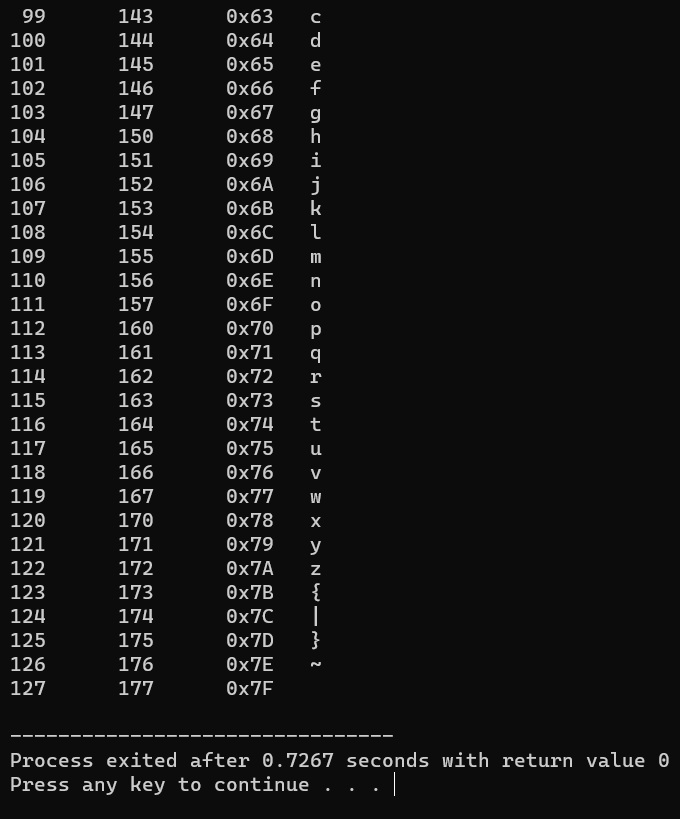
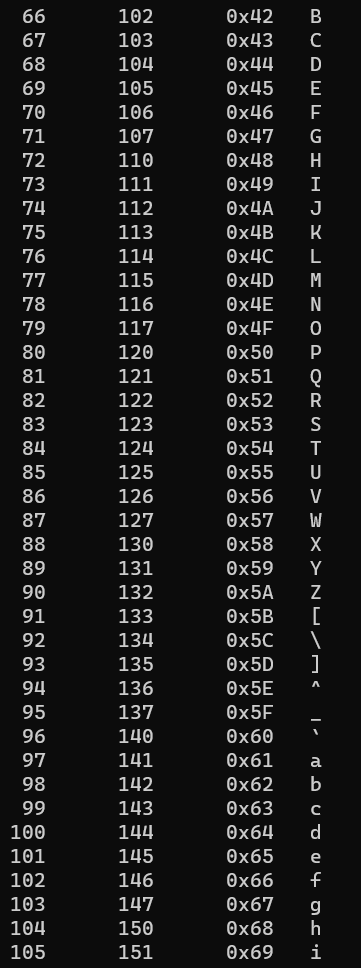
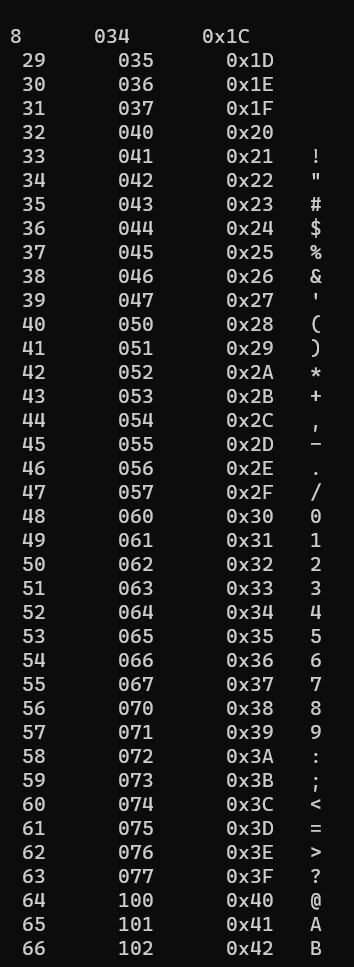
printf("%3d %03o 0x%02X %c\n", i, i, i, (char)i);

}

return 0;

}





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

#include <stdio.h>

int main() {

double base, result = 1;

int exponent;

printf("Program By Krish Grover:\n");

printf("Enter the base: ");

scanf("%lf", &base);

printf("Enter the exponent: ");

scanf("%d", &exponent);

if (exponent < 0) {

printf("Exponent should be a non-negative integer.\n");

} else {

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

result \*= base;

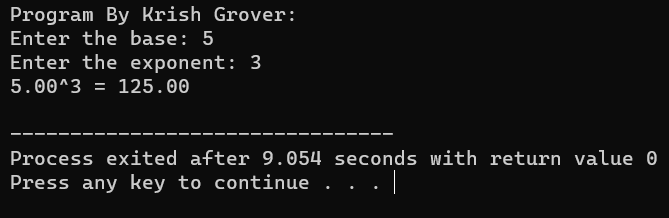
}

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

}

return 0;

}



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

#include <stdio.h>

int main() {

int number;

printf("Program By Krish Grover:\n");

printf("Enter a number: ");

scanf("%d", &number);

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

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

if (number % i == 0) {

printf("%d ", i);

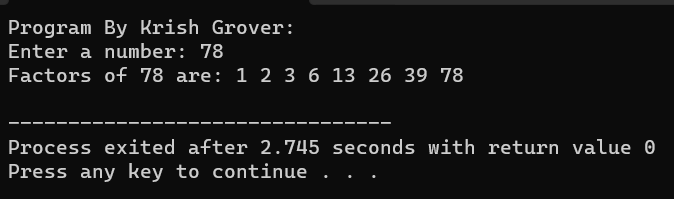
}

}

printf("\n");

return 0;

}



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

#include <stdio.h>

int main() {

int number;

unsigned long long factorial = 1;

printf("Program By Krish Grover:\n");

printf("Enter a non-negative integer: ");

scanf("%d", &number);

if (number < 0) {

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

} else {

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

factorial \*= i;

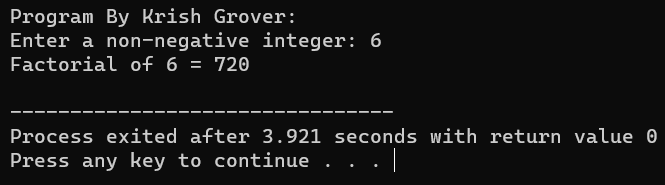
}

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

}

return 0;

}



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

#include <stdio.h>

// Function to find the GCD using the Euclidean algorithm

int findGCD(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

int main() {

int num1, num2;

printf("Program By Krish Grover:\n");

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

scanf("%d", &num2);

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

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

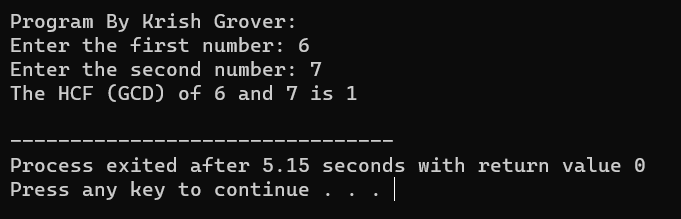
} else {

int gcd = findGCD(num1, num2);

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

}

}



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

#include <stdio.h>

// Function to find the GCD using the Euclidean algorithm

int findGCD(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

// 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("Program By Krish Grover:\n");

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

scanf("%d", &num2);

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

printf("LCM is not defined for non-positive numbers.\n");

} else {

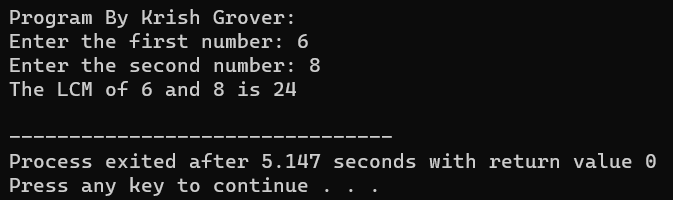
int lcm = findLCM(num1, num2);

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

}

return 0;

}



1. 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 number) {

if (number <= 1) {

return false;

}

if (number <= 3) {

return true;

}

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

return false;

}

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

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

return false;

}

}

return true;

}

int main() {

int num;

printf("Program By Krish Grover:\n");

printf("Enter a positive integer: ");

scanf("%d", &num);

if (isPrime(num)) {

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

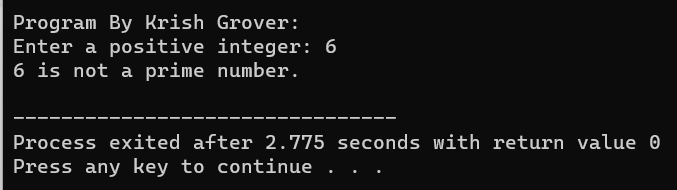
} else {

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

}

return 0;

}



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

#include <stdio.h>

// Function to check if a number is prime

bool isPrime(int num) {

if (num <= 1) {

return false;

}

if (num <= 3) {

return true;

}

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

return false;

}

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

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

return false;

}

}

return true;

}

int main() {

int n;

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

scanf("%d", &n);

if (n < 2) {

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

} else {

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

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

if (isPrime(i)) {

printf("%d, ", i);

}

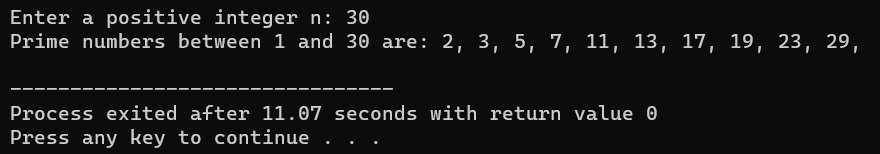
}

printf("\n");

}

return 0;

}



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

#include <stdio.h>

#include <stdbool.h>

// Function to check if a number is prime

bool isPrime(int num) {

if (num <= 1) {

return false;

}

if (num <= 3) {

return true;

}

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

return false;

}

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

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

return false;

}

}

return true;

}

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

if (n < 2) {

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

} else {

long long sum = 0; // Use long long to handle large sums

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

if (isPrime(i)) {

sum += i;

}

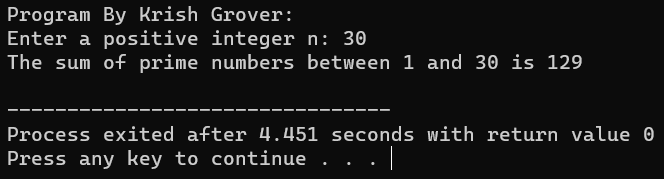
}

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

}

return 0;

}



1. 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 a prime

}

if (num <= 3) {

return 1; // Prime

}

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

return 0; // Not a prime

}

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

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

return 0; // Not a prime

}

}

return 1; // Prime

}

// Function to find and print prime factors

void findPrimeFactors(int num) {

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

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

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

while (num % i == 0) {

printf("%d ", i);

num /= i;

}

}

}

printf("\n");

}

int main() {

int n;

printf("Program By Krish Grover:\n");

printf("Enter a positive integer: ");

scanf("%d", &n);

if (n < 2) {

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

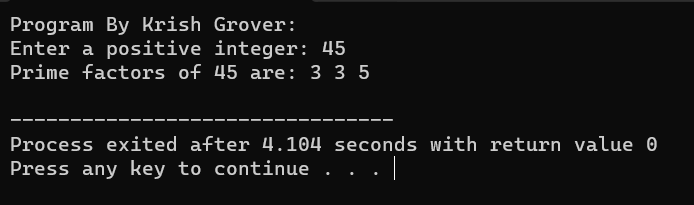
} else {

findPrimeFactors(n);

}

return 0;

}



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

#include <stdio.h>

#include <math.h>

// Function to calculate the number of digits in a number

int countDigits(int num) {

int count = 0;

while (num != 0) {

num /= 10;

count++;

}

return count;

}

// Function to check if a number is an Armstrong number

int isArmstrong(int num) {

int originalNum = num;

int numDigits = countDigits(num);

int sum = 0;

while (num != 0) {

int digit = num % 10;

sum += pow(digit, numDigits);

num /= 10;

}

return (sum == originalNum);

}

int main() {

int number;

printf("Program By Krish Grover:\n");

printf("Enter a number: ");

scanf("%d", &number);

if (isArmstrong(number)) {

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

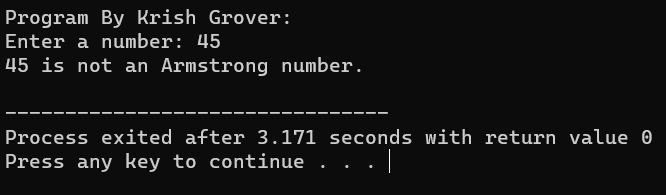
} else {

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

}

return 0;

}



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

#include <stdio.h>

#include <math.h>

// Function to calculate the number of digits in a number

int countDigits(int num) {

int count = 0;

while (num != 0) {

num /= 10;

count++;

}

return count;

}

// Function to check if a number is an Armstrong number

int isArmstrong(int num) {

int originalNum = num;

int numDigits = countDigits(num);

int sum = 0;

while (num != 0) {

int digit = num % 10;

sum += pow(digit, numDigits);

num /= 10;

}

return (sum == originalNum);

}

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

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

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

if (isArmstrong(i)) {

printf("%d, ", i);

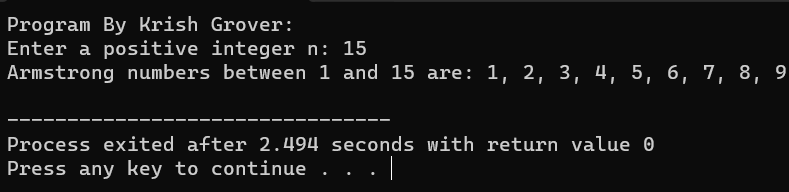
}

}

printf("\n");

return 0;

}



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

#include <stdio.h>

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

int isPerfect(int num) {

int sum = 1; // 1 is always a divisor

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

if (num % i == 0) {

if (i \* i == num) {

sum += i;

} else {

sum += i + (num / i);

}

}

}

return (sum == num);

}

int main() {

int number;

printf("Program By Krish Grover:\n");

printf("Enter a positive integer: ");

scanf("%d", &number);

if (isPerfect(number)) {

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

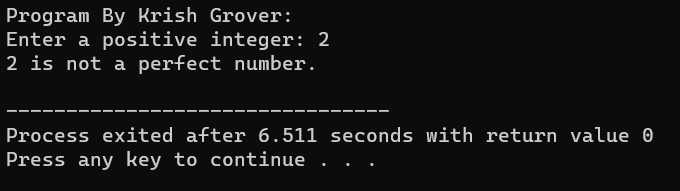
} else {

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

}

return 0;

}



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

#include <stdio.h>

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

int isPerfect(int num) {

int sum = 1; // 1 is always a divisor

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

if (num % i == 0) {

if (i \* i == num) {

sum += i;

} else {

sum += i + (num / i);

}

}

}

return (sum == num);

}

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

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

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

if (isPerfect(i)) {

printf("%d, ", i);

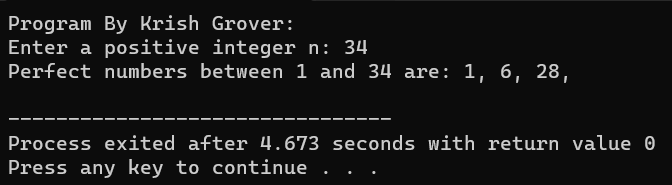
}

}

printf("\n");

return 0;

}



1. 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) {

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

return 1;

}

return num \* factorial(num - 1);

}

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

int isStrong(int num) {

int originalNum = num;

int sum = 0;

while (num != 0) {

int digit = num % 10;

sum += factorial(digit);

num /= 10;

}

return (sum == originalNum);

}

int main() {

int number;

printf("Program By Krish Grover:\n");

printf("Enter a positive integer: ");

scanf("%d", &number);

if (isStrong(number)) {

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

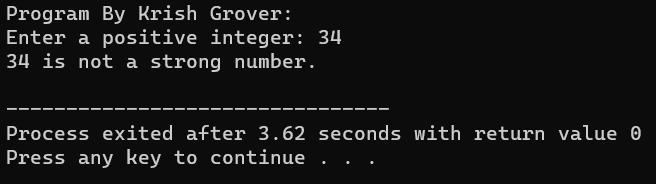
} else {

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

}

return 0;

}



1. 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) {

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

return 1;

}

return num \* factorial(num - 1);

}

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

int isStrong(int num) {

int originalNum = num;

int sum = 0;

while (num != 0) {

int digit = num % 10;

sum += factorial(digit);

num /= 10;

}

return (sum == originalNum);

}

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

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

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

if (isStrong(i)) {

printf("%d, ", i);

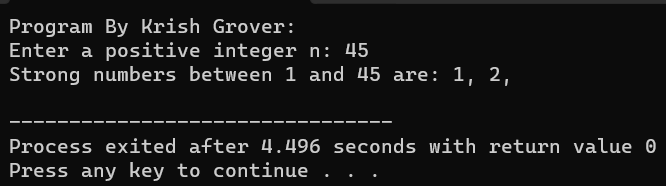
}

}

printf("\n");

return 0;

}



1. 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("Program By Krish Grover:\n");

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

scanf("%d", &n);

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

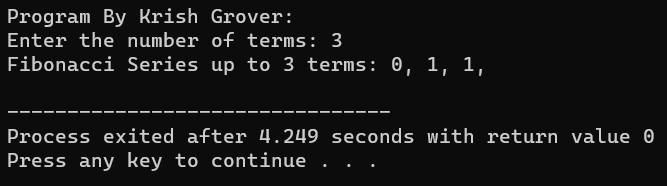
}

}

printf("\n");

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to find the one's complement of a binary number

void onesComplement(char binary[]) {

int length = strlen(binary);

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

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

binary[i] = '1';

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

binary[i] = '0';

}

}

}

int main() {

char binary[100];

printf("Program By Krish Grover:\n");

printf("Enter a binary number: ");

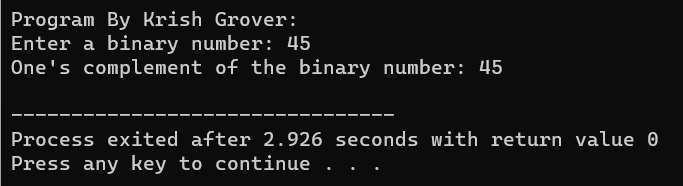
scanf("%s", binary);

onesComplement(binary);

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

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to find the one's complement of a binary number

void onesComplement(char binary[]) {

int length = strlen(binary);

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

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

binary[i] = '1';

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

binary[i] = '0';

}

}

}

// Function to find the two's complement of a binary number

void twosComplement(char binary[]) {

onesComplement(binary);

int length = strlen(binary);

int carry = 1;

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

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

binary[i] = '1';

carry = 0;

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

binary[i] = '0';

}

}

}

int main() {

char binary[100];

printf("Program By Krish Grover:\n");

printf("Enter a binary number: ");

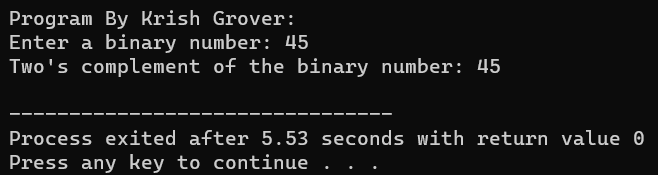
scanf("%s", binary);

twosComplement(binary);

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

return 0;

}



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

#include <stdio.h>

int main()

{

long int binarynum, octalnum = 0, j = 1, remainder;

printf("Program By Krish Grover:\n");

printf("Enter the value for binary number: ");

scanf("%ld", &binarynum);

while (binarynum != 0)

{

remainder = binarynum % 10;

octalnum = octalnum + remainder \* j;

j = j \* 2;

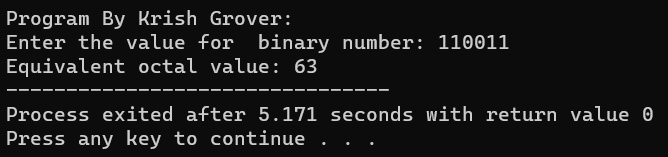
binarynum = binarynum / 10;

}

printf("Equivalent octal value: %lo", octalnum);

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert binary to decimal

int binaryToDecimal(char binary[]) {

int binaryLength = strlen(binary);

int decimal = 0;

int position = 1;

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

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

decimal += position;

}

position \*= 2;

}

return decimal;

}

int main() {

char binary[100];

printf("Program By Krish Grover:\n");

printf("Enter a binary number: ");

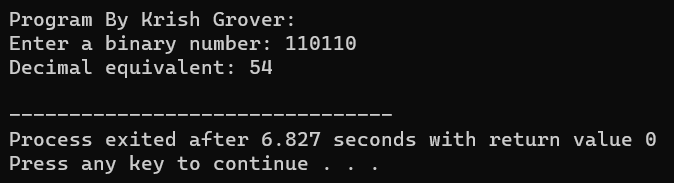
scanf("%s", binary);

int decimal = binaryToDecimal(binary);

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

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert binary to hexadecimal

void binaryToHexadecimal(char binary[]) {

int binaryLength = strlen(binary);

int position = 0;

char hexadecimal[50]; // To store the hexadecimal digits

int hexadecimalValue = 0;

int exponent = 0;

// Pad the binary number with leading zeros to ensure groups of 4 bits

int padding = 4 - (binaryLength % 4);

if (padding != 4) {

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

binary[binaryLength + i] = '0';

}

binaryLength += padding;

}

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

int bit = binary[i] - '0';

hexadecimalValue += bit \* (1 << exponent);

exponent++;

if (exponent == 4 || i == 0) {

// Convert 4 bits to a hexadecimal digit

char hexDigit;

if (hexadecimalValue < 10) {

hexDigit = '0' + hexadecimalValue;

} else {

hexDigit = 'A' + hexadecimalValue - 10;

}

// Store the hexadecimal digit

hexadecimal[position++] = hexDigit;

// Reset values for the next group of 4 bits

hexadecimalValue = 0;

exponent = 0;

}

}

printf("Hexadecimal equivalent: 0x");

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

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

}

printf("\n");

}

int main() {

char binary[100];

printf("Program By Krish Grover:\n");

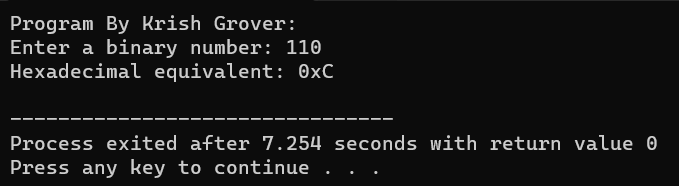
printf("Enter a binary number: ");

scanf("%s", binary);

binaryToHexadecimal(binary);

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert an octal digit to binary

char\* octalDigitToBinary(char octalDigit) {

switch (octalDigit) {

case '0': return "000";

case '1': return "001";

case '2': return "010";

case '3': return "011";

case '4': return "100";

case '5': return "101";

case '6': return "110";

case '7': return "111";

default: return NULL;

}

}

// Function to convert octal to binary

void octalToBinary(char octal[]) {

int octalLength = strlen(octal);

char binary[3 \* octalLength + 1]; // Each octal digit converts to 3 binary digits

int binaryPosition = 0;

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

char octalDigit = octal[i];

char\* binaryDigits = octalDigitToBinary(octalDigit);

if (binaryDigits) {

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

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

}

} else {

printf("Invalid octal digit: %c\n", octalDigit);

return;

}

}

binary[binaryPosition] = '\0';

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

}

int main() {

char octal[100];

printf("Program By Krish Grover:\n");

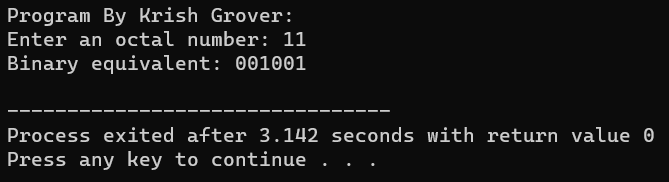
printf("Enter an octal number: ");

scanf("%s", octal);

octalToBinary(octal);

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert octal to decimal

int octalToDecimal(char octal[]) {

int octalLength = strlen(octal);

int decimal = 0;

int position = 1;

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

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

decimal += octalDigit \* position;

position \*= 8;

}

return decimal;

}

int main() {

char octal[100];

printf("Program By Krish Grover:\n");

printf("Enter an octal number: ");

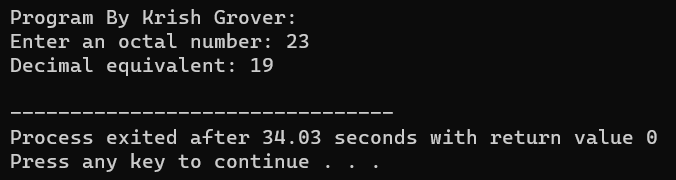
scanf("%s", octal);

int decimal = octalToDecimal(octal);

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

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert octal to hexadecimal

void octalToHexadecimal(char octal[]) {

int octalLength = strlen(octal);

char binary[4 \* octalLength + 1]; // Each octal digit converts to 3 binary digits

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

int binaryPosition = 0;

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

char octalDigit = octal[i];

int octalValue = octalDigit - '0';

// Convert the octal digit to 3 binary digits

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

binary[binaryPosition++] = ((octalValue >> j) & 1) + '0';

}

}

binary[binaryPosition] = '\0';

// Now, convert the binary representation to hexadecimal

int binaryLength = strlen(binary);

int remainder = binaryLength % 4;

int padding = (remainder == 0) ? 0 : 4 - remainder;

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

binary[binaryLength + i] = '0';

}

binaryLength += padding;

int hexIndex = 0;

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

int hexValue = 0;

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

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

}

if (hexValue < 10) {

hexadecimal[hexIndex++] = hexValue + '0';

} else {

hexadecimal[hexIndex++] = hexValue - 10 + 'A';

}

}

hexadecimal[hexIndex] = '\0';

printf("Hexadecimal equivalent: 0x%s\n", hexadecimal);

}

int main() {

char octal[100];

printf("Program By Krish Grover:\n");

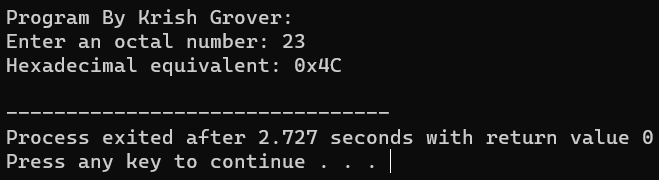
printf("Enter an octal number: ");

scanf("%s", octal);

octalToHexadecimal(octal);

return 0;

}



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

#include <stdio.h>

// Function to convert decimal to binary

void decimalToBinary(int decimal) {

if (decimal == 0) {

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

return;

}

int binary[32]; // Assuming a 32-bit binary representation

int index = 0;

while (decimal > 0) {

binary[index] = decimal % 2;

decimal /= 2;

index++;

}

printf("Binary equivalent: ");

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

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

}

printf("\n");

}

int main() {

int decimal;

printf("Program By Krish Grover:\n");

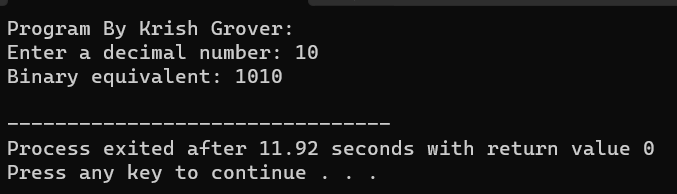
printf("Enter a decimal number: ");

scanf("%d", &decimal);

decimalToBinary(decimal);

return 0;

}



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

#include <stdio.h>

// Function to convert decimal to octal

void decimalToOctal(int decimal) {

if (decimal == 0) {

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

return;

}

int octal[32]; // Assuming a 32-bit octal representation

int index = 0;

while (decimal > 0) {

octal[index] = decimal % 8;

decimal /= 8;

index++;

}

printf("Octal equivalent: ");

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

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

}

printf("\n");

}

int main() {

int decimal;

printf("Program By Krish Grover:\n");

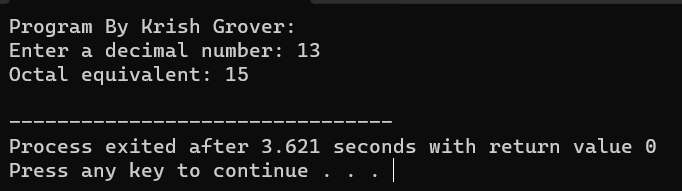
printf("Enter a decimal number: ");

scanf("%d", &decimal);

decimalToOctal(decimal);

return 0;

}



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

#include <stdio.h>

// Function to convert decimal to hexadecimal

void decimalToHexadecimal(int decimal) {

if (decimal == 0) {

printf("Hexadecimal equivalent: 0x0\n");

return;

}

char hexadecimal[32]; // Assuming a 32-bit hexadecimal representation

int index = 0;

while (decimal > 0) {

int remainder = decimal % 16;

if (remainder < 10) {

hexadecimal[index] = '0' + remainder;

} else {

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

}

decimal /= 16;

index++;

}

printf("Hexadecimal equivalent: 0x");

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

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

}

printf("\n");

}

int main() {

int decimal;

printf("Program By Krish Grover:\n");

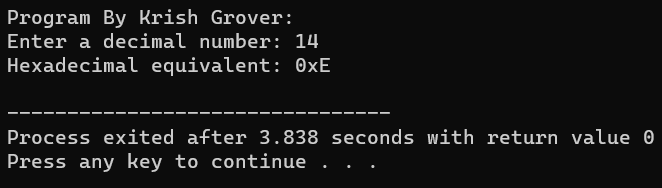
printf("Enter a decimal number: ");

scanf("%d", &decimal);

decimalToHexadecimal(decimal);

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert a hexadecimal digit to binary

char\* hexDigitToBinary(char hexDigit) {

switch (hexDigit) {

case '0': return "0000";

case '1': return "0001";

case '2': return "0010";

case '3': return "0011";

case '4': return "0100";

case '5': return "0101";

case '6': return "0110";

case '7': return "0111";

case '8': return "1000";

case '9': return "1001";

case 'A':

case 'a': return "1010";

case 'B':

case 'b': return "1011";

case 'C':

case 'c': return "1100";

case 'D':

case 'd': return "1101";

case 'E':

case 'e': return "1110";

case 'F':

case 'f': return "1111";

default: return NULL;

}

}

// Function to convert hexadecimal to binary

void hexadecimalToBinary(char hexadecimal[]) {

int hexLength = strlen(hexadecimal);

char binary[4 \* hexLength + 1]; // Each hex digit converts to 4 binary digits

int binaryPosition = 0;

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

char hexDigit = hexadecimal[i];

char\* binaryDigits = hexDigitToBinary(hexDigit);

if (binaryDigits) {

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

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

}

} else {

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

return;

}

}

binary[binaryPosition] = '\0';

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

}

int main() {

char hexadecimal[100];

printf("Program By Krish Grover:\n");

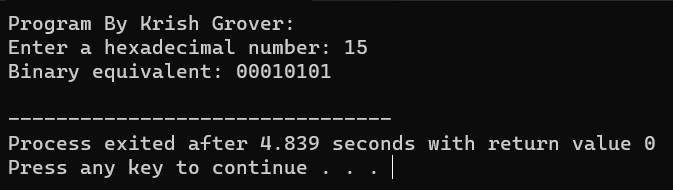
printf("Enter a hexadecimal number: ");

scanf("%s", hexadecimal);

hexadecimalToBinary(hexadecimal);

return 0;

}



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

#include <stdio.h>

#include <string.h>

// Function to convert a hexadecimal digit to octal

char\* hexDigitToOctal(char hexDigit) {

switch (hexDigit) {

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";

case '8': return "1000";

case '9': return "1001";

case 'A':

case 'a': return "1010";

case 'B':

case 'b': return "1011";

case 'C':

case 'c': return "1100";

case 'D':

case 'd': return "1101";

case 'E':

case 'e': return "1110";

case 'F':

case 'f': return "1111";

default: return NULL;

}

}

// Function to convert hexadecimal to octal

void hexadecimalToOctal(char hexadecimal[]) {

int hexLength = strlen(hexadecimal);

char binary[4 \* hexLength + 1]; // Each hex digit converts to 4 binary digits

char octal[3 \* hexLength + 1]; // Each group of 3 binary digits converts to 1 octal digit

int binaryPosition = 0;

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

char hexDigit = hexadecimal[i];

char\* binaryDigits = hexDigitToOctal(hexDigit);

if (binaryDigits) {

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

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

}

} else {

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

return;

}

}

binary[binaryPosition] = '\0';

// Pad the binary number with leading zeros to ensure groups of 3 bits

int padding = 3 - (binaryPosition % 3);

if (padding != 3) {

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

binary[binaryPosition + i] = '0';

}

binaryPosition += padding;

}

binary[binaryPosition] = '\0';

int binaryLength = strlen(binary);

int octalPosition = 0;

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

int octalValue = 0;

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

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

}

octal[octalPosition++] = octalValue + '0';

}

octal[octalPosition] = '\0';

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

}

int main() {

char hexadecimal[100];

printf("Program By Krish Grover:\n");

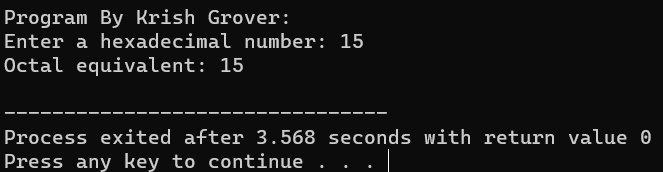
printf("Enter a hexadecimal number: ");

scanf("%s", hexadecimal);

hexadecimalToOctal(hexadecimal);

return 0;

}



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

#include <stdio.h>

#include <string.h>

#include <math.h>

// Function to convert hexadecimal to decimal

int hexadecimalToDecimal(char hexadecimal[]) {

int hexLength = strlen(hexadecimal);

int decimal = 0;

int position = 0;

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

char hexDigit = hexadecimal[i];

int hexValue;

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

hexValue = hexDigit - '0';

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

hexValue = hexDigit - 'A' + 10;

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

hexValue = hexDigit - 'a' + 10;

} else {

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

return -1;

}

decimal += hexValue \* pow(16, position);

position++;

}

return decimal;

}

int main() {

char hexadecimal[100];

printf("Program By Krish Grover:\n");

printf("Enter a hexadecimal number: ");

scanf("%s", hexadecimal);

int decimal = hexadecimalToDecimal(hexadecimal);

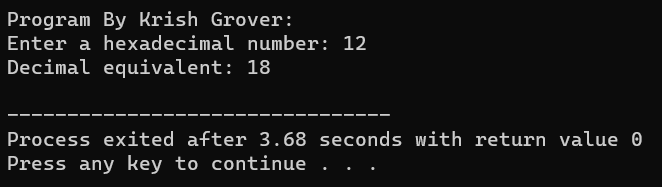
if (decimal != -1) {

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

}

return 0;

}



Pattern Exercises

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

**(i)**

#include <stdio.h>

int main() {

int rows;

printf("Program By Krish Grover:\n");

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

scanf("%d", &rows);

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

// Print spaces before stars

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

printf(" ");

}

// Print stars

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

printf("\*");

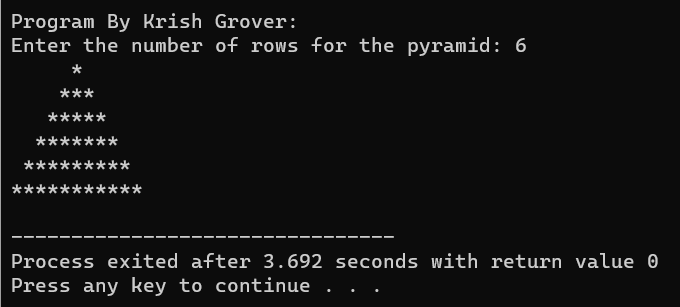
}

printf("\n");

}

return 0;

}



(ii)

#include <stdio.h>

int main()

{

printf("Program By Krish Grover:\n");

int i, space, n = 5, j = 0;

for (i = 0; i < n - 1; i++) {

// second for loop is used to print spaces

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

printf(" ");

}

for (j = 0; j <= 2 \* i; j++) {

if (j == 0 || j == 2 \* i)

printf("\*");

else

printf(" ");

}

// print the new line after every row

printf("\n");

}

// used to print last row

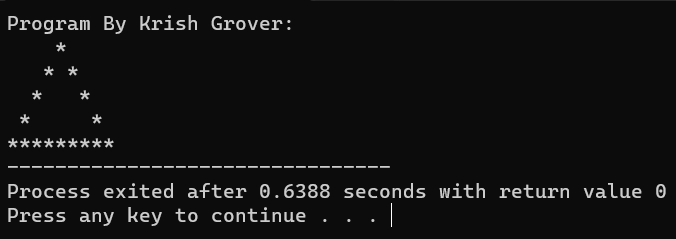
for (i = 0; i < 2 \* n - 1; i++) {

printf("\*");

}

return 0;

}



(iii)

#include <stdio.h>

int main()

{

printf("Program By Krish Grover:\n");

int rows = 8, i, j, space;

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

{

// Loop to print the blank space

for (space = 0;

space < rows - i; ++space)

printf(" ");

// Loop to print the half of

// the star triangle

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

printf("\* ");

// Loop to print the rest of

// the star triangle

for (j = 0; j < i - 1; ++j)

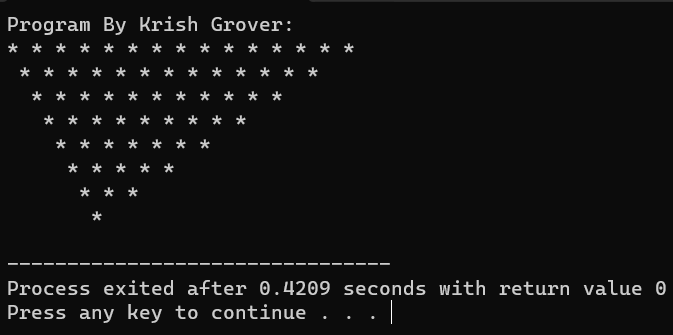
printf("\* ");

printf("\n");

}

return 0;

}



(iv)

// C program to print an inverted hollow

// star pyramid using for loop

#include <stdio.h>

void pattern\_fun(int row)

{

// To iterate through the rows

for (int j = 1; j <= row; j++)

{

// To print the beginning spaces

for (int sp = 1; sp <= j - 1; sp++)

{

printf(" ");

}

// Iterating from jth column to

// last column (row\*2 - (2\*j - 1))

int last\_col = (row \* 2 - (2 \* j - 1));

// To iterate through column

for (int k = 1; k <= last\_col; k++)

{

// To Print all star for first

// row (j==1) jth column (k==1)

// and for last column

// (row\*2 - (2\*j - 1))

if (j == 1 || k == 1)

printf("\*");

else if (k == last\_col)

printf("\*");

else

printf(" ");

}

// Proceeding to next row.

printf("\n");

}

}

int main()

{

printf("Program By Krish Grover:\n");

// Number of rows

int row = 7;

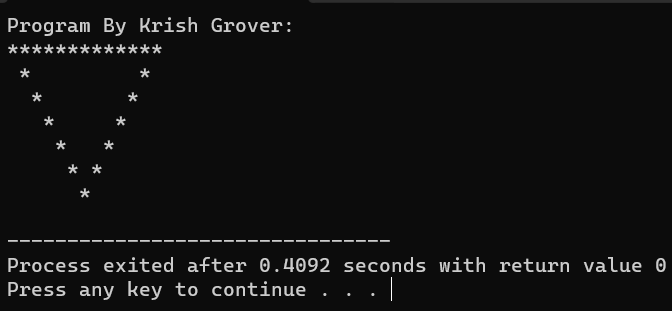
// Calling the function to

// print the pattern.

pattern\_fun(row);

return 0;

}



(v)

#include<stdio.h>

int main()

{

int i, j, N, columns;

/\* Input number of columns from user \*/

printf("Enter 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)

{

/\* Increment number of columns per row for upper part \*/

columns++;

}

else

{

/\* Decrement number of columns per row for lower part \*/

columns--;

}

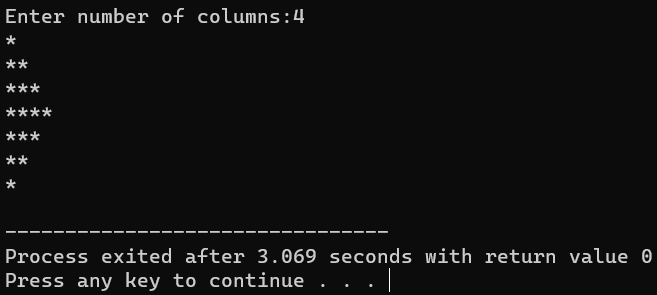
/\* Move to next line \*/

printf("\n");

}

return 0;

}



(vi)

#include <stdio.h>

int main()

{

int i, j, N;

int star, spaces;

printf("Program By Krish Grover:\n");

/\* Input number of columns to print from user \*/

printf("Enter number of columns : ");

scanf("%d", &N);

spaces = N-1;

star = 1;

/\* Iterate through rows \*/

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

{

/\* Print leading spaces \*/

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

printf(" ");

/\* Print stars \*/

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

printf("\*");

/\* Move to next line \*/

printf("\n");

if(i < N)

{

star++;

spaces--;

}

else

{

star--;

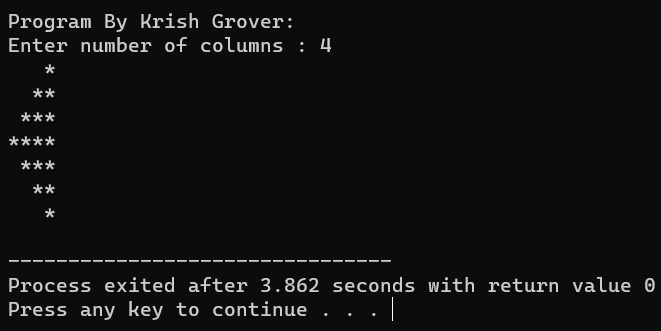
spaces++;

}

}

return 0;

}



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

(i)

#include <stdio.h>

int main()

{

int rows, cols, i, j;

printf("Program By Krish Grover:\n");

/\* Input rows and columns from user \*/

printf("Enter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

/\* Iterate through rows \*/

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

{

/\* Iterate through columns \*/

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

{

printf("1");

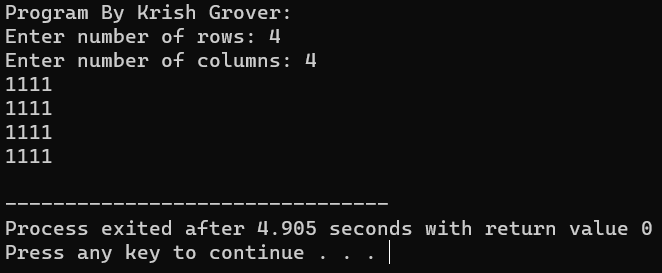
}

printf("\n");

}

return 0;

}



(ii)

#include <stdio.h>

int main()

{

int rows, cols, i, j;

printf("Program By Krish Grover:\n");

/\* Input rows and columns from user \*/

printf("Enter 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++)

{

// Print 1 if current row is odd

if(i%2 == 1)

{

printf("1");

}

else

{

printf("0");

}

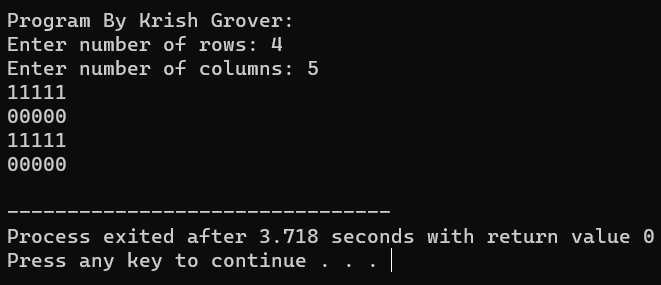
}

printf("\n");

}

return 0;

}



(iii)

#include <stdio.h>

int main()

{

int rows, cols, i, j;

printf("Program By Krish Grover:\n");

/\* Input rows and columns from user \*/

printf("Enter 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++)

{

// Print 1 if current column is even

if(j%2 == 1)

{

printf("0");

}

else

{

printf("1");

}

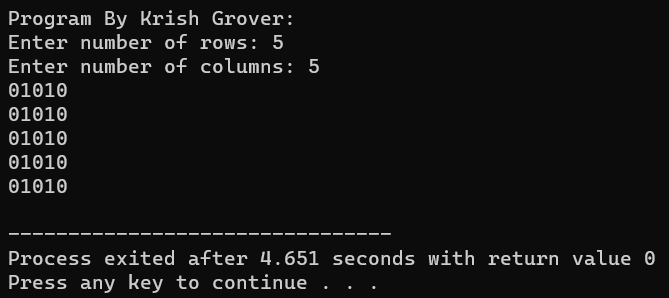
}

printf("\n");

}

return 0;

}



(iv)

#include <stdio.h>

int main()

{

int rows, cols, i, j;

printf("Program By Krish Grover:\n");

/\* Input rows and columns from user \*/

printf("Enter 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++)

{

/\*

\* Print 1 if its first or last row

\* Print 1 if its first or last column

\*/

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

{

printf("1");

}

else

{

printf("0");

}

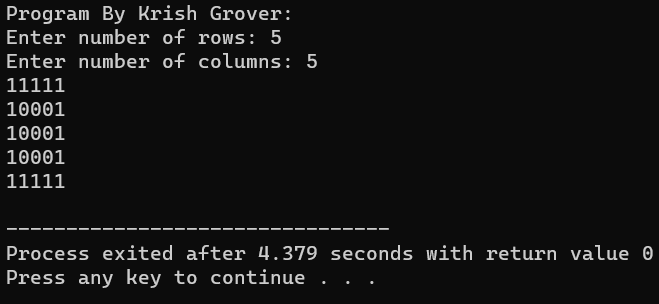
}

printf("\n");

}

return 0;

}



(v)

#include <stdio.h>

int main()

{

int rows, cols, i, j;

int centerRow, centerCol;

printf("Program By Krish Grover:\n");

/\* Input rows and columns from user \*/

printf("Enter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

/\* Find center row and column \*/

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");

}

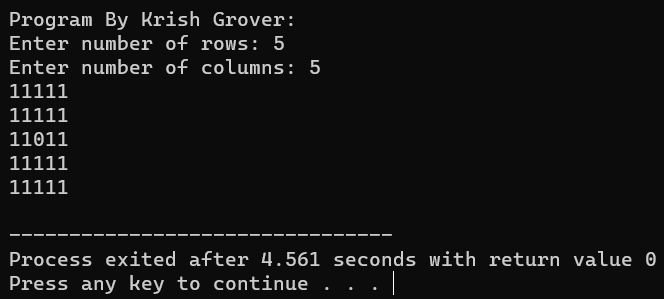
}

printf("\n");

}

return 0;

}



(vi)

#include <stdio.h>

int main()

{

int rows, cols, i, j, k;

printf("Program By Krish Grover:\n");

/\* Input rows and columns from user \*/

printf("Enter 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");

}

// If k = 1 then k \*= -1 => -1

// If k = -1 then k \*= -1 => 1

k \*= -1;

}

if(cols % 2 == 0)

{

k \*= -1;

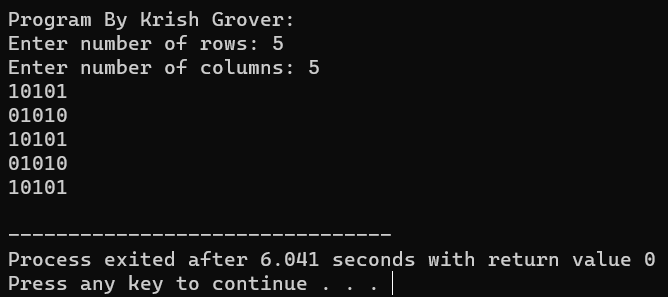
}

printf("\n");

}

return 0;

}



If…Else Exercises

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

#include <stdio.h>

int main()

{

int num1, num2;

printf("Program By Krish Grover:\n");

/\* Input two numbers from user \*/

printf("Enter two numbers: ");

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

/\* If num1 is maximum \*/

if(num1 > num2)

{

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

}

/\* If num2 is maximum \*/

if(num2 > num1)

{

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

}

/\* Additional condition check for equality \*/

if(num1 == num2)

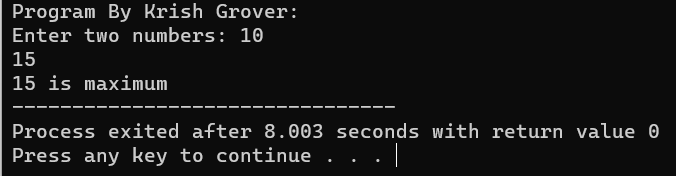
{

printf("Both are equal");

}

return 0;

}



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

#include <stdio.h>

int main()

{

int num1, num2, num3, max;

printf("Program By Krish Grover:\n");

/\* Input three numbers from user \*/

printf("Enter three numbers: ");

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

if(num1 > num2)

{

if(num1 > num3)

{

/\* If num1 > num2 and num1 > num3 \*/

max = num1;

}

else

{

/\* If num1 > num2 but num1 > num3 is not true \*/

max = num3;

}

}

else

{

if(num2 > num3)

{

/\* If num1 is not > num2 and num2 > num3 \*/

max = num2;

}

else

{

/\* If num1 is not > num2 and num2 > num3 \*/

max = num3;

}

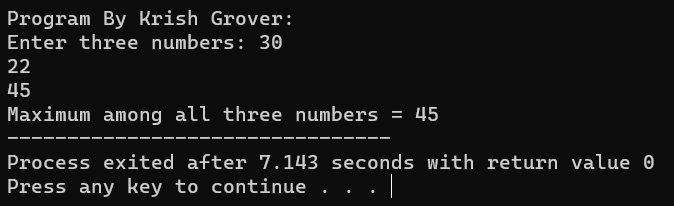
}

/\* Print maximum value \*/

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()

{

int num;

printf("Program By Krish Grover:\n");

/\* Input number from user \*/

printf("Enter any number: ");

scanf("%d", &num);

if(num > 0)

{

printf("Number is POSITIVE");

}

if(num < 0)

{

printf("Number is NEGATIVE");

}

if(num == 0)

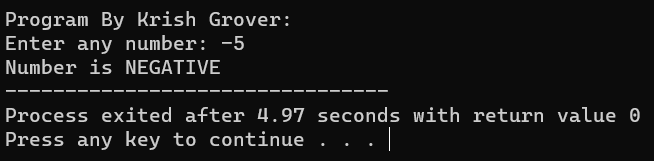
{

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()

{

int num;

printf("Program By Krish Grover:\n");

/\* Input number from user \*/

printf("Enter any number: ");

scanf("%d", &num);

/\*

\* If num modulo division 5 is 0

\* and num modulo division 11 is 0 then

\* the number is divisible by 5 and 11 both

\*/

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

{

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

}

else

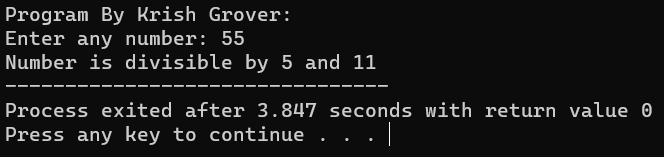
{

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()

{

int num;

printf("Program By Krish Grover:\n");

/\* Input number from user \*/

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

scanf("%d", &num);

/\* Check if the number is divisible by 2 then it is even \*/

if(num % 2 == 0)

{

/\* num % 2 is 0 \*/

printf("Number is Even.");

}

else

{

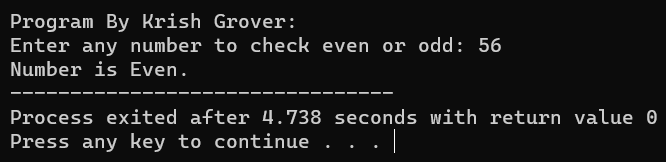
/\* num % 2 is 1 \*/

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()

{

int year;

printf("Program By Krish Grover:\n");

/\* Input year from user \*/

printf("Enter year : ");

scanf("%d", &year);

/\*

\* If year is exactly divisible by 4 and year is not divisible by 100

\* or year is exactly divisible by 400 then

\* the year is leap year.

\* Else year is normal year

\*/

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

{

printf("LEAP YEAR");

}

else

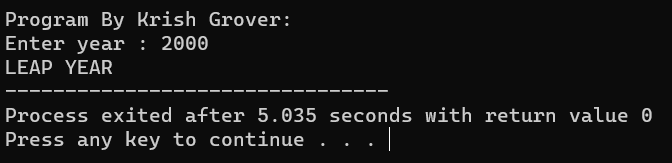
{

printf("COMMON YEAR");

}

return 0;

}



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

#include<stdio.h>

int main()

{

char ch;

printf("Program By Krish Grover:\n");

/\* Input a character from user \*/

printf("Enter any character: ");

scanf("%c", &ch);

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

{

printf("Character is an ALPHABET.");

}

else

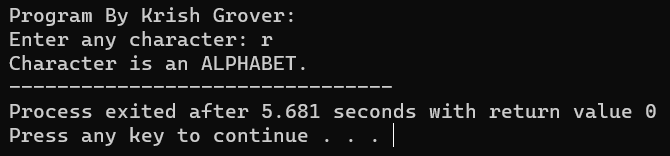
{

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()

{

char ch;

printf("Program By Krish Grover:\n");

/\* Input character from user \*/

printf("Enter any character: ");

scanf("%c", &ch);

/\* Alphabet check \*/

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

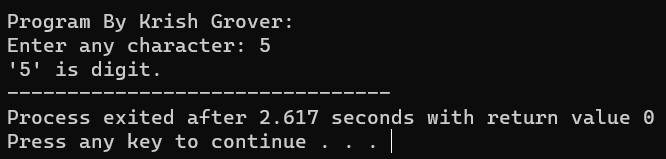
{

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

}

return 0;

}



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

#include <stdio.h>

int main()

{

char ch;

printf("Program By Krish Grover:\n");

/\* Input character from user \*/

printf("Enter 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

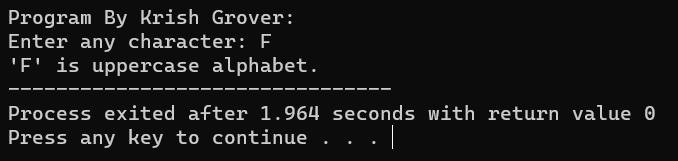
{

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

}

return 0;

}



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

#include <stdio.h>

int main()

{

char ch;

printf("Program By Krish Grover:\n");

/\* Input character from user \*/

printf("Enter 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

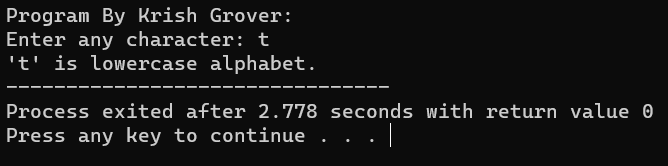
{

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()

{

int week;

printf("Program By Krish Grover:\n");

/\* Input week number from user \*/

printf("Enter week 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

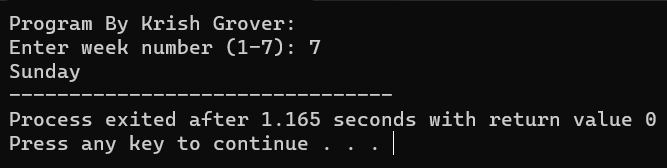
{

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()

{

int month;

printf("Program By Krish Grover:\n");

/\* Input month number from user \*/

printf("Enter 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

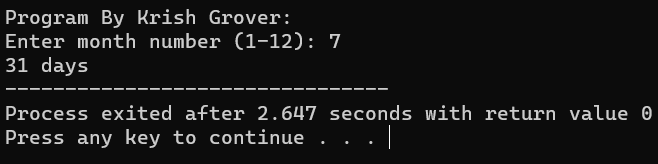
{

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()

{

int amount;

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

printf("Program By Krish Grover:\n");

/\* Initialize all notes to 0 \*/

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

/\* Input amount from user \*/

printf("Enter 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;

}/\* Print required notes \*/

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);

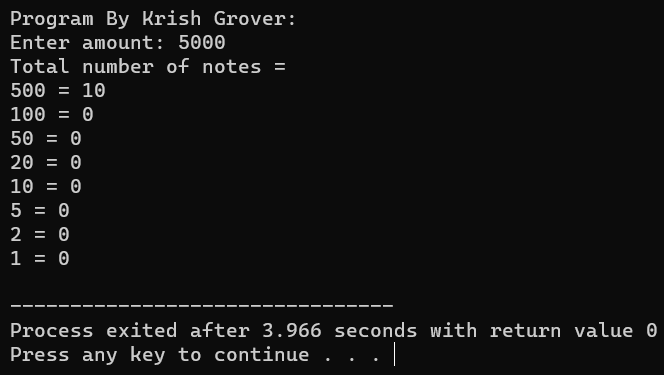
printf("10 = %d\n", note10);

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

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

printf("1 = %d\n", note1);return 0;

}



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, sum;

printf("Program By Krish Grover:\n");

/\* Input all three angles of triangle \*/

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

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

/\* Calculate sum of angles \*/

sum = angle1 + angle2 + angle3;

/\*

\* If sum of angles is 180 and

\* angle1, angle2, angle3 is not 0 then

\* triangle is valid.

\*/

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

{

printf("Triangle is valid.");

}

else

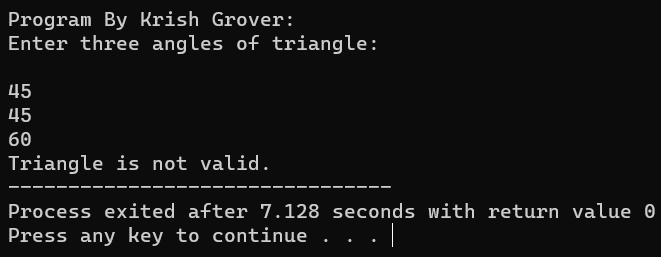
{

printf("Triangle is not valid.");

}

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()

{

int side1, side2, side3;

printf("Program By Krish Grover:\n");

/\* Input three sides of a triangle \*/

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

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

if((side1 + side2) > side3)

{

if((side2 + side3) > side1)

{

if((side1 + side3) > side2)

{

/\*

\* If side1 + side2 > side3 and

\* side2 + side3 > side1 and

\* side1 + side3 > side2 then

\* the triangle is valid.

\*/

printf("Triangle is valid.");

}

else

{

printf("Triangle is not valid.");

}

}

else

{

printf("Triangle is not valid.");

}

}

else

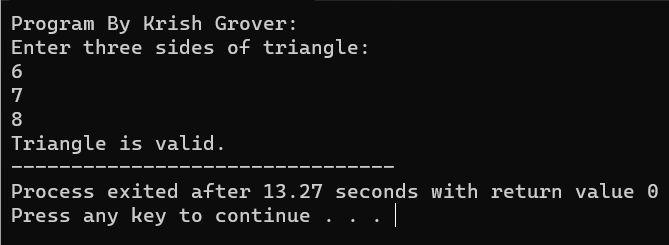
{

printf("Triangle is not valid.");

}

return 0;

}



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

#include <stdio.h>

int main()

{

int side1, side2, side3;

printf("Program By Krish Grover:\n");

/\* Input sides of a triangle \*/

printf("Enter three sides of triangle: ");

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

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

{

/\* If all sides are equal \*/

printf("Equilateral triangle.");

}

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

{

/\* If any two sides are equal \*/

printf("Isosceles triangle.");

}

else

{

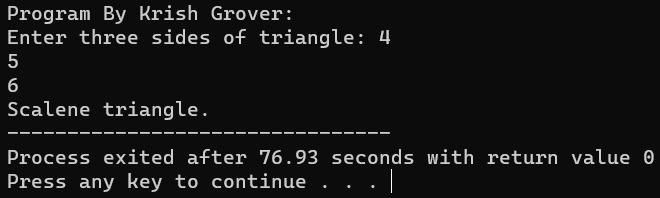
/\* If none sides are equal \*/

printf("Scalene triangle.");

}

return 0;

}



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

#include <stdio.h>

#include <math.h> /\* Used for sqrt() \*/

int main()

{

float a, b, c;

float root1, root2, imaginary;

float discriminant;

printf("Program By Krish Grover:\n");

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

scanf("%f%f%f", &a, &b, &c);

/\* Find discriminant of the equation \*/

discriminant = (b \* b) - (4 \* a \* c);

/\* Find the nature of discriminant \*/

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);

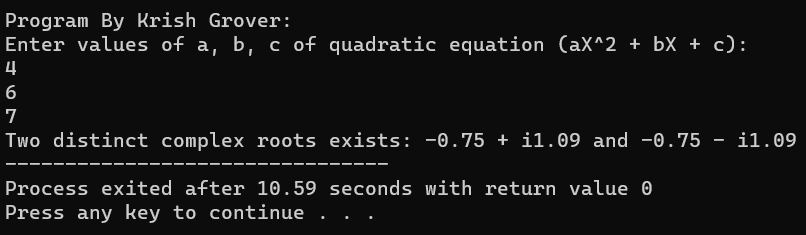
printf("Two distinct complex roots exists: %.2f + i%.2f and %.2f - i%.2f",

root1, imaginary, root2, imaginary);

}

return 0;

}



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

#include <stdio.h>

int main()

{

int cp,sp, amt;

printf("Program By Krish Grover:\n");

/\* Input cost price and selling price of a product \*/

printf("Enter cost price: ");

scanf("%d", &cp);

printf("Enter selling price: ");

scanf("%d", &sp);

if(sp > cp)

{

/\* Calculate Profit \*/

amt = sp - cp;

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

}

else if(cp > sp)

{

/\* Calculate Loss \*/

amt = cp - sp;

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

}

else

{

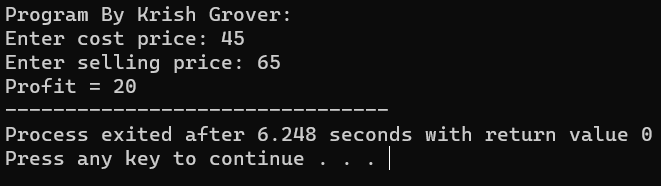
/\* Neither profit nor loss \*/

printf("No Profit No Loss.");

}

return 0;

}



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()

{

int phy, chem, bio, math, comp;

float per;

printf("Program By Krish Grover:\n");

/\* Input marks of five subjects from user \*/

printf("Enter five subjects marks: ");

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

/\* Calculate percentage \*/

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

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

/\* Find grade according to the percentage \*/

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

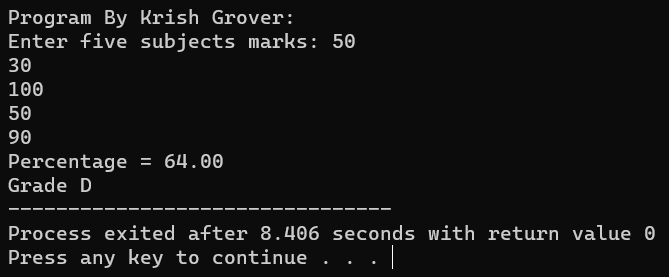
{

printf("Grade F");

}

return 0;

}



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, gross, da, hra;

printf("Program By Krish Grover:\n");

/\* Input basic salary of employee \*/

printf("Enter basic salary of an employee: ");

scanf("%f", &basic);

/\* Calculate D.A and H.R.A according to specified conditions \*/

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;

}

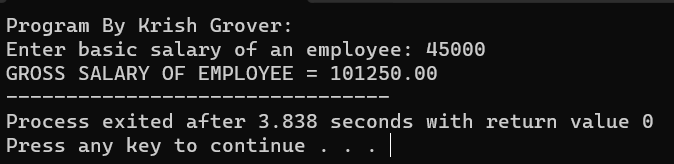
/\* Calculate gross salary \*/

gross = basic + hra + da;

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

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()

{

int unit;

float amt, total\_amt, sur\_charge;

printf("Program By Krish Grover:\n");

/\* Input unit consumed from user \*/

printf("Enter total units consumed: ");

scanf("%d", &unit);

/\* Calculate electricity bill according to given conditions \*/

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);

}

/\*

\* Calculate total electricity bill

\* after adding surcharge

\*/

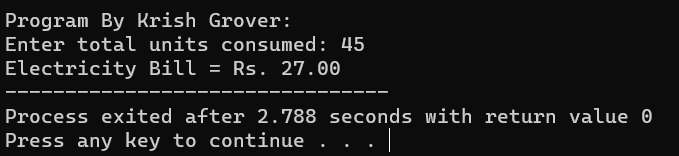
sur\_charge = amt \* 0.20;

total\_amt = amt + sur\_charge;

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

return 0;

}



Array Exercises

1. Write a program in C to read n number of values in an array and display them in reverse order.

#include <stdio.h>

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

if (n <= 0) {

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

return 1;

}

int arr[n];

printf("Enter %d values:\n", n);

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

scanf("%d", &arr[i]);

}

printf("Values in reverse order:\n");

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

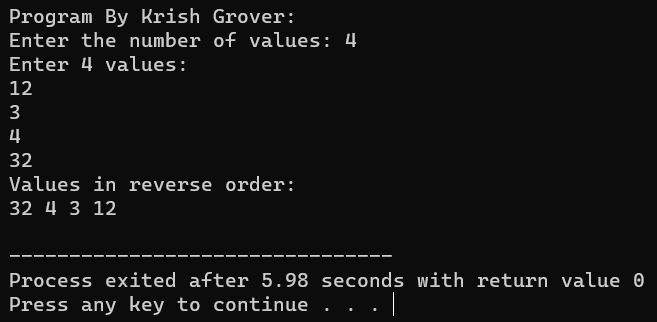
printf("%d ", arr[i]);

}

printf("\n");

return 0;

}



1. Write a program in C to find the sum of all elements of the array.

#include <stdio.h>

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

if (n <= 0) {

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

return 1;

}

int arr[n];

int sum = 0;

printf("Enter %d values:\n", n);

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

scanf("%d", &arr[i]);

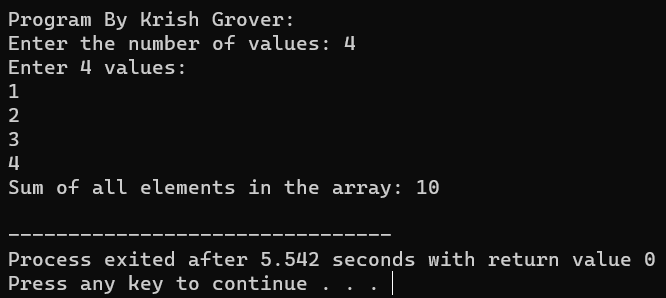
sum += arr[i]; // Add the current element to the running sum

}

printf("Sum of all elements in the array: %d\n", sum);

return 0;

}



1. Write a program in C to copy the elements of one array into another array.

#include <stdio.h>

int main() {

int n;

printf("Program By Krish Grover:\n");

printf("Enter the number of elements in the source array: ");

scanf("%d", &n);

if (n <= 0) {

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

return 1;

}

int sourceArray[n];

int destinationArray[n];

printf("Enter %d elements for the source array:\n", n);

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

scanf("%d", &sourceArray[i]);

}

// Copy elements from sourceArray to destinationArray

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

destinationArray[i] = sourceArray[i];

}

printf("Elements in the destination array after copying:\n");

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

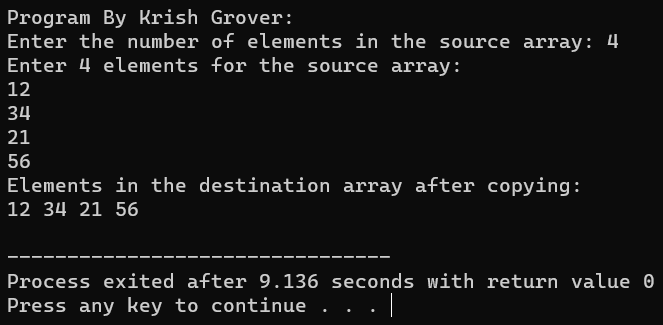
printf("%d ", destinationArray[i]);

}

printf("\n");

return 0;

}



1. Write a program in C to count the total number of duplicate elements in an array.

#include <stdio.h>

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

if (n <= 0) {

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

return 1;

}

int arr[n];

printf("Enter %d elements for the array:\n", n);

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

scanf("%d", &arr[i]);

}

int count = 0; // Initialize a variable to count duplicate elements

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

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

if (arr[i] == arr[j]) {

count++;

break; // Break the inner loop when a duplicate is found

}

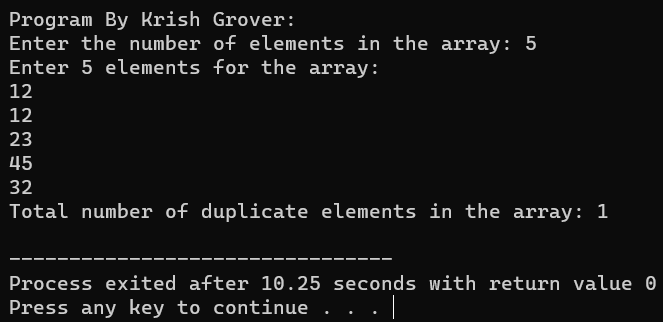
}

}

printf("Total number of duplicate elements in the array: %d\n", count);

return 0;

}



1. Write a program in C to find the maximum and minimum elements in an array.

#include <stdio.h>

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

if (n <= 0) {

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

return 1;

}

int arr[n];

printf("Enter %d elements for the array:\n", n);

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

scanf("%d", &arr[i]);

}

if (n == 1) {

printf("Maximum and minimum element is %d\n", arr[0]);

return 0;

}

int max = arr[0];

int min = arr[0];

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

if (arr[i] > max) {

max = arr[i];

}

if (arr[i] < min) {

min = arr[i];

}

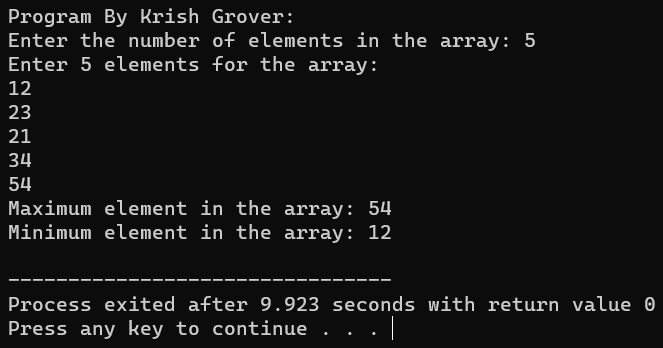
}

printf("Maximum element in the array: %d\n", max);

printf("Minimum element in the array: %d\n", min);

return 0;

}



1. Write a C program to sort the elements of an array in descending order.

#include <stdio.h>

void selectionSort(int arr[], int n) {

for (int i = 0; i < n - 1; i++) {

int maxIndex = i;

// Find the index of the maximum element in the unsorted part of the array

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

if (arr[j] > arr[maxIndex]) {

maxIndex = j;

}

}

// Swap the maximum element with the element at index 'i'

int temp = arr[i];

arr[i] = arr[maxIndex];

arr[maxIndex] = temp;

}

}

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

if (n <= 0) {

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

return 1;

}

int arr[n];

printf("Enter %d elements for the array:\n", n);

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

scanf("%d", &arr[i]);

}

selectionSort(arr, n);

printf("Array sorted in descending order:\n");

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

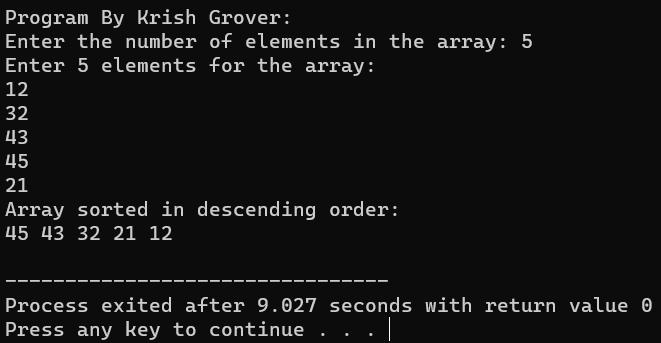
printf("%d ", arr[i]);

}

printf("\n");

return 0;

}



1. Write a program in C to separate odd and even integers into separate arrays.

#include <stdio.h>

int main() {

int n;

printf("Program By Krish Grover:\n");

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

scanf("%d", &n);

if (n <= 0) {

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

return 1;

}

int arr[n];

int evenArr[n]; // Array to store even integers

int oddArr[n]; // Array to store odd integers

int evenCount = 0; // Count of even integers

int oddCount = 0; // Count of odd integers

printf("Enter %d elements for the array:\n", n);

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

scanf("%d", &arr[i]);

if (arr[i] % 2 == 0) {

evenArr[evenCount] = arr[i];

evenCount++;

} else {

oddArr[oddCount] = arr[i];

oddCount++;

}

}

printf("Even integers in the array: ");

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

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

}

printf("\n");

printf("Odd integers in the array: ");

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

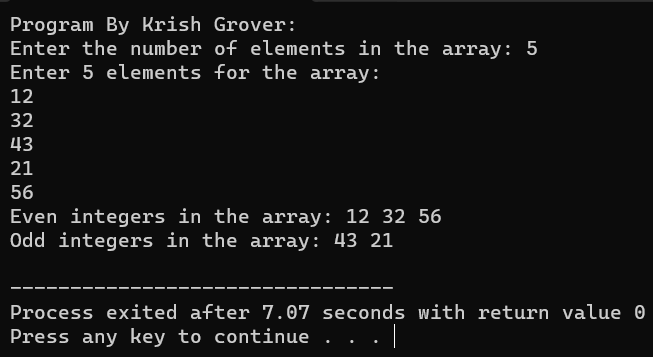
printf("%d ", oddArr[i]);

}

printf("\n");

return 0;

}



1. Write a program in C to merge two arrays of the same size sorted in descending/ascending order.

#include <stdio.h>

void mergeSortedArrays(int arr1[], int arr2[], int mergedArr[], int n) {

int i = 0, j = 0, k = 0;

while (i < n && j < n) {

if (arr1[i] <= arr2[j]) {

mergedArr[k] = arr1[i];

i++;

} else {

mergedArr[k] = arr2[j];

j++;

}

k++;

}

// Copy the remaining elements from arr1, if any

while (i < n) {

mergedArr[k] = arr1[i];

i++;

k++;

}

// Copy the remaining elements from arr2, if any

while (j < n) {

mergedArr[k] = arr2[j];

j++;

k++;

}

}

int main() {

int n;

printf("Program By Krish Grover:\n");

printf("Enter the number of elements in each array: ");

scanf("%d", &n);

if (n <= 0) {

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

return 1;

}

int arr1[n];

int arr2[n];

int mergedArr[2 \* n];

printf("Enter %d elements for the first array in ascending order:\n", n);

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

scanf("%d", &arr1[i]);

}

printf("Enter %d elements for the second array in ascending order:\n", n);

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

scanf("%d", &arr2[i]);

}

mergeSortedArrays(arr1, arr2, mergedArr, n);

printf("Merged array in ascending order:\n");

for (int i = 0; i < 2 \* n; i++) {

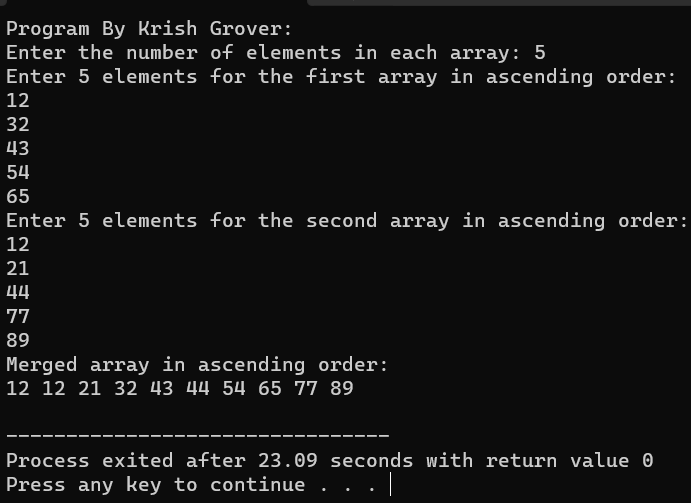
printf("%d ", mergedArr[i]);

}

printf("\n");

return 0;

}



1. Write a program in C to merge two arrays of the same size sorted in descending order.

#include <stdio.h>

void mergeSortedArrays(int arr1[], int arr2[], int mergedArr[], int n) {

int i = 0, j = 0, k = 0;

while (i < n && j < n) {

if (arr1[i] >= arr2[j]) { // Modified merge condition for descending order

mergedArr[k] = arr1[i];

i++;

} else {

mergedArr[k] = arr2[j];

j++;

}

k++;

}

// Copy the remaining elements from arr1, if any

while (i < n) {

mergedArr[k] = arr1[i];

i++;

k++;

}

// Copy the remaining elements from arr2, if any

while (j < n) {

mergedArr[k] = arr2[j];

j++;

k++;

}

}

int main() {

int n;

printf("Program By Krish Grover:\n");

printf("Enter the number of elements in each array: ");

scanf("%d", &n);

if (n <= 0) {

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

return 1;

}

int arr1[n];

int arr2[n];

int mergedArr[2 \* n];

printf("Enter %d elements for the first array in descending order:\n", n);

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

scanf("%d", &arr1[i]);

}

printf("Enter %d elements for the second array in descending order:\n", n);

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

scanf("%d", &arr2[i]);

}

mergeSortedArrays(arr1, arr2, mergedArr, n);

printf("Merged array in descending order:\n");

for (int i = 0; i < 2 \* n; i++) {

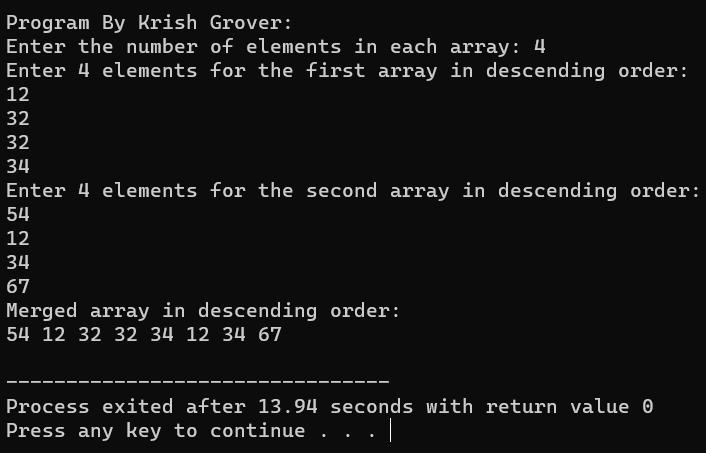
printf("%d ", mergedArr[i]);

}

printf("\n");

return 0;

}



1. WAP using Switch case: Consider two matrices of the size m and n. Implement matrix operation and display. Show these things in program 1) Read matrix elements and display 2) Matrix Multiplication and display 3) addition of matrix and display 4)Subtraction of Matrix and display 5)Transpose of Matrix and display.

#include <stdio.h>

void readMatrix(int matrix[][10], int m, int n) {

printf("Enter matrix elements:\n");

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

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

scanf("%d", &matrix[i][j]);

}

}

}

void displayMatrix(int matrix[][10], int m, int n) {

printf("Matrix:\n");

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

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

printf("%d\t", matrix[i][j]);

}

printf("\n");

}

}

void matrixMultiplication(int A[][10], int m, int n, int B[][10], int p, int q, int result[][10]) {

if (n != p) {

printf("Matrix multiplication not possible.\n");

return;

}

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

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

result[i][j] = 0;

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

result[i][j] += A[i][k] \* B[k][j];

}

}

}

}

void matrixAddition(int A[][10], int B[][10], int m, int n, int result[][10]) {

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

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

result[i][j] = A[i][j] + B[i][j];

}

}

}

void matrixSubtraction(int A[][10], int B[][10], int m, int n, int result[][10]) {

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

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

result[i][j] = A[i][j] - B[i][j];

}

}

}

void transposeMatrix(int matrix[][10], int m, int n, int result[][10]) {

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

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

result[j][i] = matrix[i][j];

}

}

}

int main() {

int choice, m, n, p, q;

int A[10][10], B[10][10], result[10][10];

printf("Program By Krish Grover:\n");

printf("Enter the number of rows and columns for matrix A: ");

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

printf("Enter the number of rows and columns for matrix B: ");

scanf("%d %d", &p, &q);

if (m != p || n != q) {

printf("Matrix dimensions do not match for matrix operations.\n");

return 1;

}

printf("Matrix A:\n");

readMatrix(A, m, n);

printf("Matrix B:\n");

readMatrix(B, p, q);

printf("Choose an operation:\n");

printf("1. Display matrices\n");

printf("2. Matrix Multiplication\n");

printf("3. Matrix Addition\n");

printf("4. Matrix Subtraction\n");

printf("5. Transpose of Matrix\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

displayMatrix(A, m, n);

displayMatrix(B, p, q);

break;

case 2:

matrixMultiplication(A, m, n, B, p, q, result);

displayMatrix(result, m, q);

break;

case 3:

matrixAddition(A, B, m, n, result);

displayMatrix(result, m, n);

break;

case 4:

matrixSubtraction(A, B, m, n, result);

displayMatrix(result, m, n);

break;

case 5:

transposeMatrix(A, m, n, result);

displayMatrix(result, n, m);

break;

default:

printf("Invalid choice.\n");

break;

}

return 0;

}

