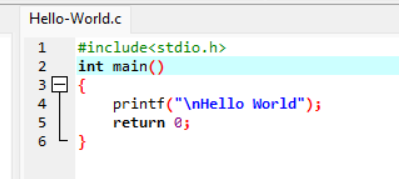
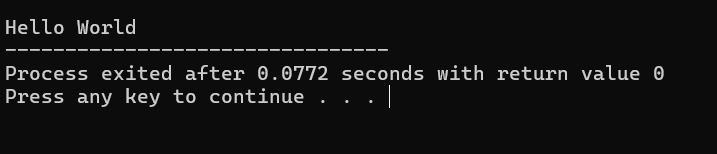
1. Research and provide three real-world applications where C programming is extensively used, such as in embedded systems, operating systems, or game development.

* Embedded Systems – Microcontroller programming, Arduino, and Raspberry Pi.
* Operating Systems – Linux, RTOS, and Free BSD.
* Game Development – Unreal Engine, Unity, and CryEngine

1. Install a C compiler on your system and configure the IDE. Write your first program to print "Hello, World!" and run it.





1. Write a C program that includes variables, constants, and comments. Declare and use different data types (int, char, float) and display their values.

#include <stdio.h>

int main() {

const float PI = 3.14159;

int age = 25;

char grade = 'A';

float height = 5.9;

printf("Age: %d\n", age);

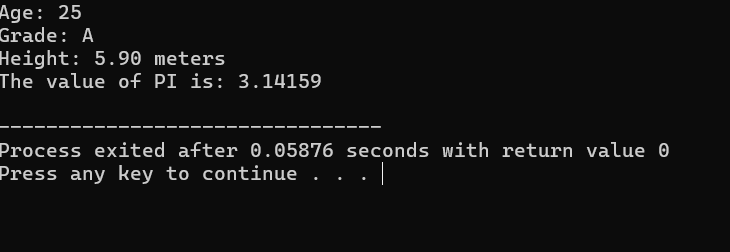
printf("Grade: %c\n", grade);

printf("Height: %.2f meters\n", height);

printf("The value of PI is: %.5f\n", PI);

return 0;

}



1. Write a C program that accepts two integers from the user and performs arithmetic, relational, and logical operations on them. Display the results.

#include <stdio.h>

int main() {

int num1, num2;

printf("Enter the first integer: ");

scanf("%d", &num1);

printf("Enter the second integer: ");

scanf("%d", &num2);

printf("\nArithmetic Operations:\n");

printf("Addition: %d + %d = %d\n", num1, num2, num1 + num2);

printf("Subtraction: %d - %d = %d\n", num1, num2, num1 - num2);

printf("Multiplication: %d \* %d = %d\n", num1, num2, num1 \* num2);

printf("Division: %d / %d = %d\n", num1, num2, num1 / num2);

printf("Modulus: %d %% %d = %d\n", num1, num2, num1 % num2);

printf("\nRelational Operations:\n");

printf("Is %d equal to %d? %d\n", num1, num2, num1 == num2);

printf("Is %d not equal to %d? %d\n", num1, num2, num1 != num2);

printf("Is %d greater than %d? %d\n", num1, num2, num1 > num2);

printf("Is %d less than %d? %d\n", num1, num2, num1 < num2);

printf("Is %d greater than or equal to %d? %d\n", num1, num2, num1 >= num2);

printf("Is %d less than or equal to %d? %d\n", num1, num2, num1 <= num2);

printf("\nLogical Operations:\n");

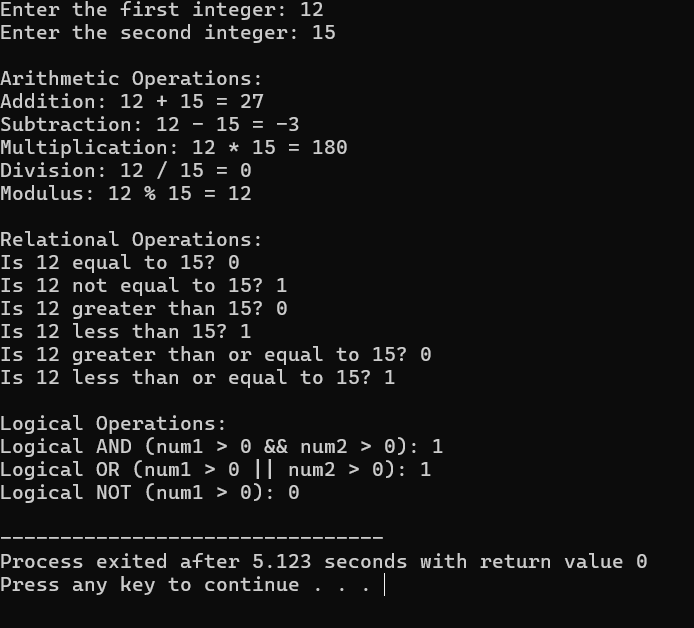
printf("Logical AND (num1 > 0 && num2 > 0): %d\n", (num1 > 0 && num2 > 0));

printf("Logical OR (num1 > 0 || num2 > 0): %d\n", (num1 > 0 || num2 > 0));

printf("Logical NOT (num1 > 0): %d\n", !(num1 > 0));

return 0;

}



1. Write a C program to check if a number is even or odd using an if-else

statement. Extend the program using a switch statement to display the month name based on the user’s input (1 for January, 2 for February, etc.).

#include <stdio.h>

int main() {

int num, month;

printf("Enter a number: ");

scanf("%d", &num);

if (num % 2 == 0) {

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

} else {

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

}

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

scanf("%d", &month);

switch (month) {

case 1:

printf("Month 1 is January.\n");

break;

case 2:

printf("Month 2 is February.\n");

break;

case 3:

printf("Month 3 is March.\n");

break;

case 4:

printf("Month 4 is April.\n");

break;

case 5:

printf("Month 5 is May.\n");

break;

case 6:

printf("Month 6 is June.\n");

break;

case 7:

printf("Month 7 is July.\n");

break;

case 8:

printf("Month 8 is August.\n");

break;

case 9:

printf("Month 9 is September.\n");

break;

case 10:

printf("Month 10 is October.\n");

break;

case 11:

printf("Month 11 is November.\n");

break;

case 12:

printf("Month 12 is December.\n");

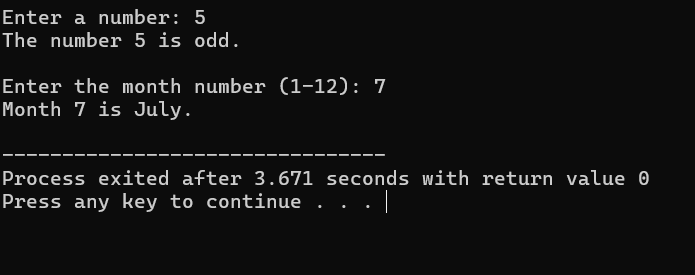
break;

default:

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

}

return 0;

}

1. Write a C program to print numbers from 1 to 10 using all three types of loops (while, for, do-while).

#include <stdio.h>

int main() {

// Using while loop

printf("Using while loop:\n");

int i = 1;

while (i <= 10) {

printf("%d ", i);

i++;

}

printf("\n");

// Using for loop

printf("Using for loop:\n");

i = 1;

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

printf("%d ", i);

}

printf("\n");

// Using do-while loop

printf("Using do-while loop:\n");

i = 1;

do {

printf("%d ", i);

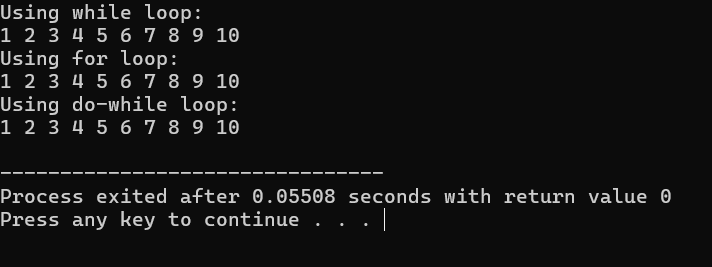
i++;

} while (i <= 10);

printf("\n");

return 0;

}



1. Write a C program that stores 5 integers in a one-dimensional array and prints them. Extend this to handle a two-dimensional array (3x3 matrix) and calculate the sum of all elements.

#include<stdio.h>

int main()

{

int a[100],i,size,sum=0;

printf("\nEnter the size of an array = ");

scanf("%d",&size);

for(i=0;i<size;i++)//scanning

{

printf("\nEnter the element on a[%d] = ",i);

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

sum = sum +a[i];//addition

}

printf("\nNormal Sequence:");//Line Break

for (i=0;i<size;i++)//printing

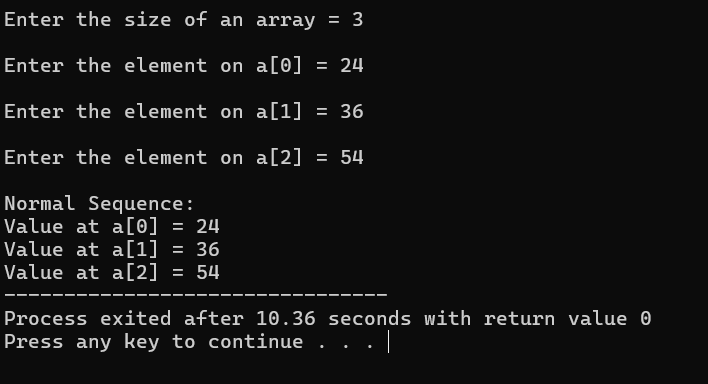
{

printf("\nValue at a[%d] = %d",i,a[i]);

}

return 0;

}



#include<stdio.h>

int main()

{

int a[10][10], b[10][10], ans[10][10];//[row][collumn] = {row \* collumn}

int i, j , k ,size;

printf("\nEnter the row and col number = ");

scanf("%d",&size);

printf("\nEnter the elements in array a = ");

int temp=1;

for(i=0;i<size;i++)

{

for(j=0;j<size;j++)

{

printf("\nElement %d = ",temp++);

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

}

}

printf("\nEnter the elements in array b = ");

for(i=0;i<size;i++)

{

for(j=0;j<size;j++)

{

printf("\nElement %d = ",temp++);

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

}

}

for(i=0;i<size;i++)

{

for(j=0;j<size;j++)

{

ans[i][j] = a[i][j] + b[i][j];

}

printf("\n");

}

printf("\nArray a = \n");

for(i=0;i<size;i++)

{

for(j=0;j<size;j++)

{

printf("%d ",a[i][j]);

}

printf("\n");

}

printf("\nArray b = \n");

for(i=0;i<size;i++)

{

for(j=0;j<size;j++)

{

printf("%d ",b[i][j]);

}

printf("\n");

}

printf("\nArray ans = \n");

for(i=0;i<size;i++)

{

for(j=0;j<size;j++)

{

printf("%d ",ans[i][j]);

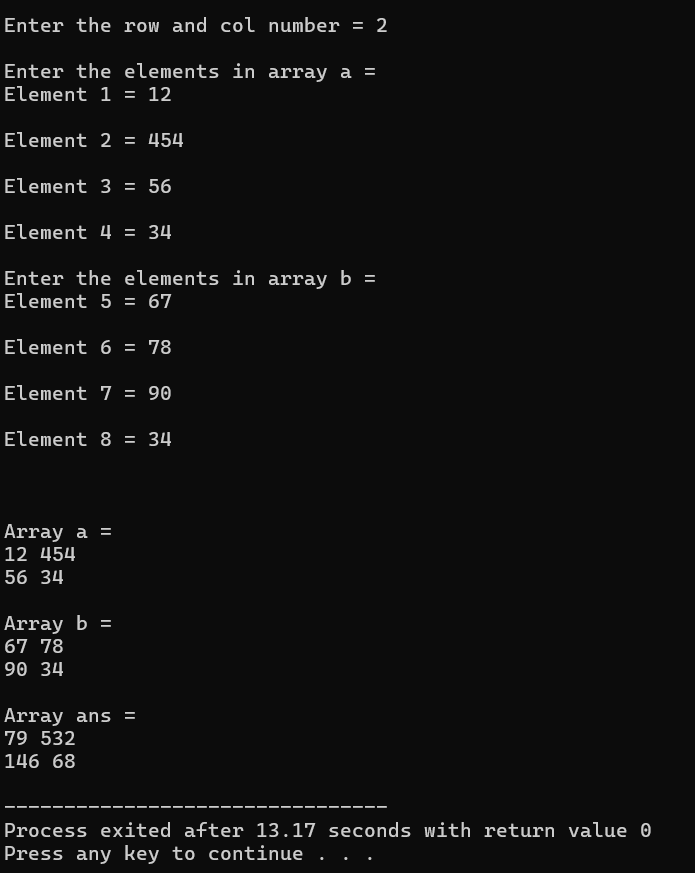
}

printf("\n");

}

return 0;

}



1. Write a C program that uses the break statement to stop printing numbers when it reaches 5. Modify the program to skip printing the number 3 using the continue statement.

#include <stdio.h>

int main() {

int i;

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

if (i == 3) {

continue; // Skip printing 3

}

if (i == 5) {

break;

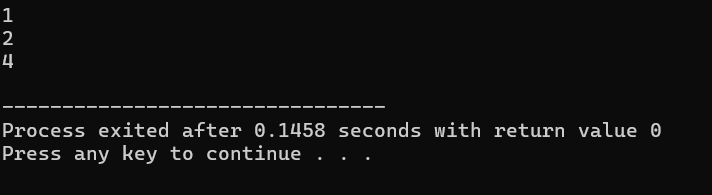
}

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

}

return 0;

}



1. Write a C program that calculates the factorial of a number using a function. Include function declaration, definition, and call.

#include<stdio.h>

int multi(int num)

{

int i, fact=1;

for(i=1;i<=num; i++)//Increment

{

fact = fact \* i;

}

printf("\nFactorial of %d is %d", num, fact);

}

int main()

{

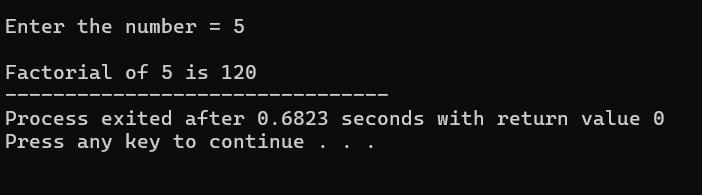
int n1;

printf("\nEnter the number = ");

scanf("%d",&n1);

multi(n1);

return 0; }



1. Write a C program to demonstrate pointer usage. Use a pointer to modify the value of a variable and print the result.

#include <stdio.h>

int main()

{

int number = 5;

int \*ptr;

ptr = &number; // Store the address of 'number' in pointer 'ptr'

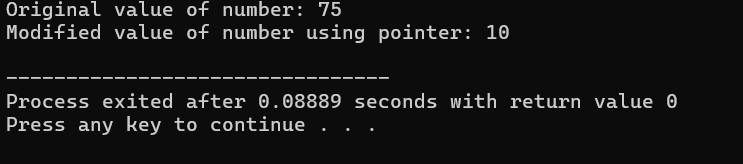
printf("Original value of number: %d\n", number);

\*ptr = 10; // Value of pointer is modified

printf("Modified value of number using pointer: %d\n", number);

return 0;

}



1. Write a C program that takes two strings from the user and concatenates them using strcat(). Display the concatenated string and its length using strlen().

#include<stdio.h>

#include<string.h>

int main()

{

char str1[100],str2[100];

printf("\nEnter the value in str1 = ");

gets(str1);

printf("\nEnter the value in str2 = ");

gets(str2);

printf("\nOriginal value of str1 = %s",str1);

printf("\nOriginal value of str2 = %s",str2);

strcat(str1,str2);

printf("\nValue of str1 after using concate function = %s",str1);

printf("\nValue of str2 after using concate function = %s",str2);

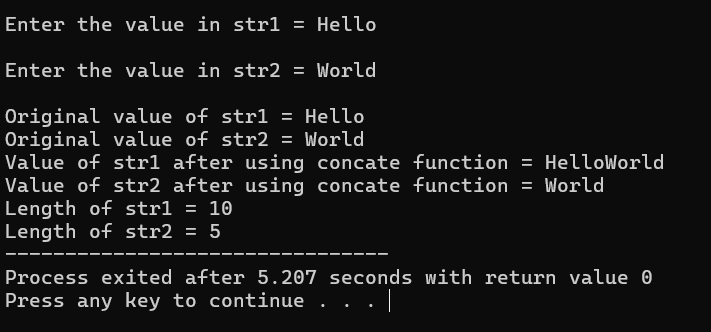
int len = strlen(str1);

printf("\nLength of str1 = %d",len);

printf("\nLength of str2 = %d",strlen(str2));

return 0;

}



1. Write a C program that defines a structure to store a student's details (name, roll number, and marks). Use an array of structures to store details of 3 students and print them.

#include<stdio.h>

#include<string.h>

struct Student

{

int id;

char name[20];

int percentage;

}s[100];

int index=0;

int main()

{

up :

printf("\n1. Add new students");

printf("\n2. Display all students");

int choice;

printf("\nEnter your choice = ");

scanf("%d",&choice);

switch(choice)

{

case 1:

addnew();

break;

case 2:

displayall();

break;

}

char select;

printf("\nPress 'Y' to continue and 'N' to exit = ");

scanf(" %c",&select);

if(select=='y' || select=='Y')

{

goto up;

}

return 0;

}

void addnew()

{

int count,i;

printf("\nEnter the count = ");

scanf("%d",&count);

for(i=0;i<count;i++)

{

printf("\nEnter the roll no. = ");

scanf("%d",&s[index].id);

printf("\nEnter the name = ");

scanf("%s",s[index].name);

printf("\nEnter the percentage = ");

scanf("%d",&s[index].percentage);

index++;

}

}

void displayall()

{

int i;

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

printf("\nID | NAME | PERCENTAGE");

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

for(i=0;i<index;i++)

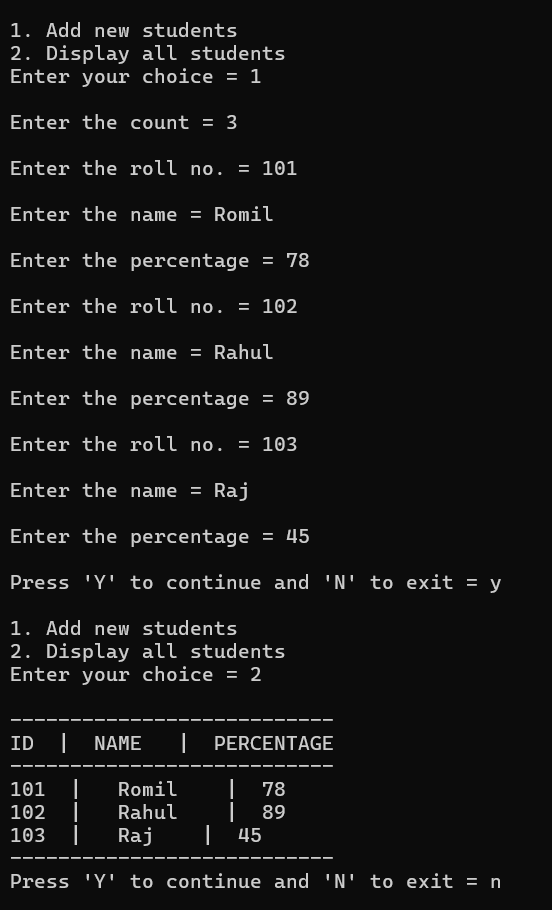
{

printf("\n%d | %s | %d",s[i].id,s[i].name,s[i].percentage);

}

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

}



1. Write a C program to create a file, write a string into it, close the file, then open the file again to read and display its contents.

#include <stdio.h>

#include <stdlib.h>

int main() {

FILE \*fp;

char str[] = "Hello, this is a file handling test!";

char ch;

// Step 1: Create and write to the file

fp = fopen("example.txt", "w"); // Open in write mode

if (fp == NULL) {

printf("File could not be created.\n");

return 1;

}

fputs(str, fp); // Write string to file

fclose(fp); // Close the file

// Step 2: Reopen and read the file

fp = fopen("second.txt","r");

if(fp==NULL)//condition

{

printf("\nFile doesn't exist");

}

else

{

while(fgets(str,sizeof(str),fp))

{

printf("%s",str);

}

}

return 0;

}

