**Practical File**

of

**Fundamentals of C Programming**

**(24CSE0107)**

**Batch-2024**

**Bachelor of Engineering (CSE)**

**A black text on a white background

Description automatically generated**

|  |  |
| --- | --- |
| **Submitted By**  **Jiya Manchanda**  2410990506  Group No 7., Sem: 2nd, BE (CSE)  Chitkara University, Punjab, 140401  India | **Submitted To**  **Dr. Chetna Sharma**  Assistant Professor, CSE  Chitkara University, Punjab, 140401  India |

**Department of**

**Computer Science and Engineering,**

**Chitkara University School of Engineering and Technology,**

**Chitkara University, Punjab, India**

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Experiment** | **Page No.** |
| 1 | Install C compiler (GCC/Code::Blocks), set up IDE, compile and run the first "Hello, World!" program. | 4-5 |
| 2 | Write a Program to show the use to input (Scanf)/output (Printf) statements and block structure of C-program by highlighting the features of "stdio.h". | 6 |
| 3 | Write a program to add two numbers and display the sum. | 7 |
| 4 | Write a program to calculate the area and the circumference of a circle by using radius as the input provided by the user. | 8 |
| 5 | Write a Program to perform addition, subtraction, division and multiplication of two numbers given as input by the user. | 9-10 |
| 6 | Write a program to evaluate each of the following equations. (i) V = u + at. (ii) S = ut+1/2at2 (iii) T=2\*a+√b+9c (iv) H=√b2+p2 | 11-12 |
| 7 | Write a program to swap two variables:  a) By using temporary variable.  b) Without using temporary variable | 13-14 |
| 8 | Write a Program to find the greatest among three numbers using:   * Conditional Operator * If-Else statement | 15-16 |
| 9 | Write the following programs using switch case statement:   * To check that an input alphabet is vowel or consonant * To check whether a number is positive, negative or zero | 17-18 |
| 10 | Write a program using while loop to print the sum of first n natural numbers. | 19 |
| 11 | Write a program to check a number is Armstrong or not using For loop. | 20 |
| 12 | Write the program to count the digits in a number and then print the reverse of the number also. | 21 |
| 13 | Write a program to generate the Fibonacci series. | 22 |
| 14 | Write a program to print the following patterns:  a)  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  \* \* \* \* \* \*  b)                 \*              \* \*           \* \* \*         \* \* \* \*      \* \* \* \* \*  \* \* \* \* \* \* | 23-24 |
| 15 | Write the program to print the following pattern:  1  2    3     4    5     6  2  4    6     8    10   12  3  6    9    12   15   18  4  8    12  16   20   24  5  10  15  20   25   30  6  12  18  24   30   36 | 25 |
| 16. | Write a program to check that the given number is prime, Armstrong or perfect using the concept of functions. | 26-27 |
| 17. | Write a program to calculate the area and circumference of a circle using functions. | 28 |
| 18. | Write a program to swap two variables using the concept of call by value and call by reference. | 29-30 |
| 19. | Write a program to perform the following operations on 1D-Array:   * Insert * Update * Delete * Display * Search | 31-35 |
| 20. | Write a program to calculate the sum of array elements by passing it to a function. | 36 |
| 21. | Write a program to show the use of passing pointer as arguments to the functions. | 37 |
| 22. | Write a program matrix multiplication using the concept of 2D array | 38-39 |
| 23. | Write a program to transpose a given matrix. | 40-41 |
| 24. | Write a program to find the factorial of a number by using the concept of recursion | 42 |
| 25. | Write a menu driven C program to show the use of in-built string functions like strlen, strcat, strcpy, strcmp, strrev etc. | 43-44 |
| 26. | Write a Program in C to display the total number of appearances of a substring provided as input by the user in a given string. | 45 |
| 27. | Write a program to display the sum of the digits of a number by using the concept of recursion. | 46 |
| 28. | Write a C program to add two distances in inch & feet using the concept of structures. | 47 |
| 29. | Write a C program to add two complex numbers using the concept of structures in C. | 48 |
| 30. | Write a program in C to store the information of five employees using both concepts i.e. array of structure and array within structure. | 49-50 |
| 31. | Write a Program in C to find and replace a specific string in a file and also display the total number of appearances of that string. | 51-52 |

**Experiment No. 1.**

**Aim:** Install C compiler (GCC/Code::Blocks), set up IDE, compile and run the first "Hello, World!" program.

**Concept Used:** Installing C compiler and setting up an IDE (VS Code in this case).  
Concepts: Compiler installation, IDE setup, writing and running a simple C program.

**Program**:

#### **Step 1: Install GCC Compiler (MinGW)**

* Download MinGW from <https://www.mingw-w64.org>.
* Run the installer and follow default installation settings.
* After installation, add the MinGW bin directory (e.g., C:\Program Files\mingw-w64\...\bin) to the **system path.**
* Open Command Prompt and type “gcc –version” to verify that GCC is installed successfully.

#### **Step 2:** **Set Up IDE (VS Code)**

* Download and install Visual Studio Code from <https://code.visualstudio.com>.
* Open VS Code and install the following extensions:
  + **C/C++** by Microsoft
  + **Code Runner** (optional, for easier execution)
* Open a new folder for your C project and create a new file named main.c.

#### **Step 3:** **Write the "Hello, World!" Program**

In main.c, write the following code:

#include <stdio.h>

int main() {

printf("Hello, World!\n");

return 0;

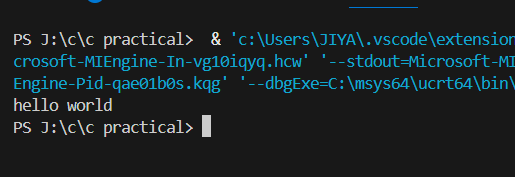
}

**Step 4:** Compile and Run the Program

Run the Program:

Click the **triangle-shaped "Run" icon** (▶️) at the top-right corner of the VS Code window. This will compile and run your program, and you should see the following output in the terminal:

**Output Screenshot:**



**Experiment No. 2.**

**Aim:** Write a Program to show the use to input (Scanf)/output (Printf) statements and block structure of C-program by highlighting the features of "stdio.h".

**Concept Used:** Basic I/O operations using scanf and printf, including stdio.h library, block structure in C.

**Program**:

#include <stdio.h>

int main()

{

int a;

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

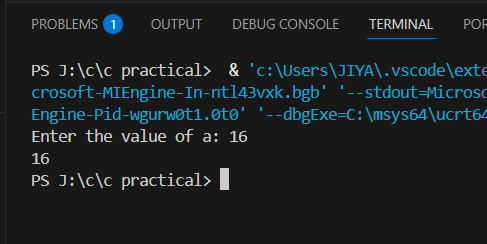
scanf("%d", &a);

printf("%d", a);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 3.**

**Aim:** Write a program to add two numbers and display the sum.

**Concept Used:**  Variable declaration, arithmetic operators, user input, and output.

**Program**:

#include <stdio.h>

int main()

{

int a,b;

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

scanf("%d", &a);

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

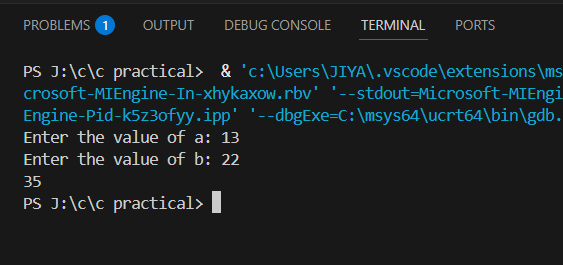
scanf("%d", &b);

printf("%d", a+b);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 4.**

**Aim:** Write a program to calculate the area and the circumference of a circle by using radius as the input provided by the user.

**Concept Used:** Mathematical operations, constants ,user input.

**Program**:

#include <stdio.h>

int main()

{

int r, circumference, area;

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

scanf("%d", &r);

circumference= (2\*3.14\*r);

area= (3.14\*r\*r);

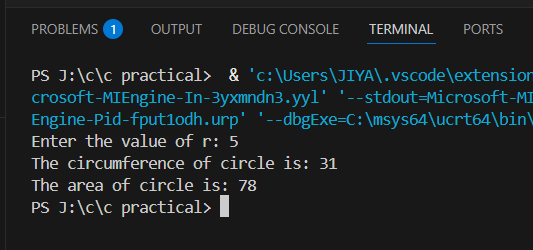
printf("The circumference of circle is: %d\n", circumference);

printf("The area of circle is: %d\n", area);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 5.**

**Aim:** Write a Program to perform addition, subtraction, division and multiplication of two numbers given as input by the user.

**Concept Used:** Arithmetic operators (+, -, \*, /), user input, and output.

**Program**:

#include <stdio.h>

int main()

{

int a,b;

char o;

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

scanf("%d", &a);

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

scanf("%d", &b);

printf("Enter the operator: ");

scanf(" %c", &o);

if (o=='+'){

printf("%d", a+b);

}

else if(o=='-'){

printf("%d", a-b);

}

else if(o=='\*'){

printf("%d", a\*b);

}

else if(o=='/'){

if(b!=0){

printf("%d", a/b);

}

else{

printf("Please enter a valid value");

}

}

else{

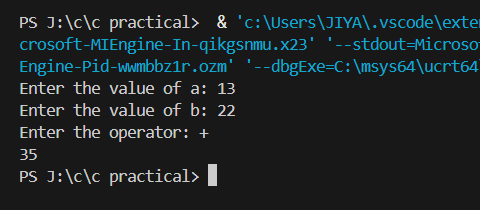
printf("Invalid operator");

}

return 0;

}

**Output Screenshot:**

****

**Experiment No. 6.**

**Aim:** Write a program to evaluate each of the following equations.  
(i) V = u + at. (ii) S = ut+1/2at2 (iii) T=2\*a+√b+9c (iv) H=√b2+p2

**Concept Used:** Variable declaration, arithmetic operations, mathematical functions, and expressions.

**Program**:

#include <stdio.h>

#include <math.h>

int main() {

float u, a, t, S, V, b, c, p, H, T;

printf("Enter values for u (initial velocity), a (acceleration), and t (time):\n");

printf("u = ");

scanf("%f", &u);

printf("a = ");

scanf("%f", &a);

printf("t = ");

scanf("%f", &t);

V = u + a \* t;

printf("\n(i) V = u + at = %.2f\n", V);

S = u \* t + 0.5 \* a \* t \* t;

printf("(ii) S = ut + 1/2 \* a \* t^2 = %.2f\n", S);

printf("\nEnter values for a, b, and c for equation (iii):\n");

printf("a = ");

scanf("%f", &a);

printf("b = ");

scanf("%f", &b);

printf("c = ");

scanf("%f", &c);

T = 2 \* a + sqrt(b + 9 \* c);

printf("(iii) T = 2\*a + sqrt(b + 9\*c) = %.2f\n", T);

printf("\nEnter values for b and p for equation (iv):\n");

printf("b = ");

scanf("%f", &b);

printf("p = ");

scanf("%f", &p);

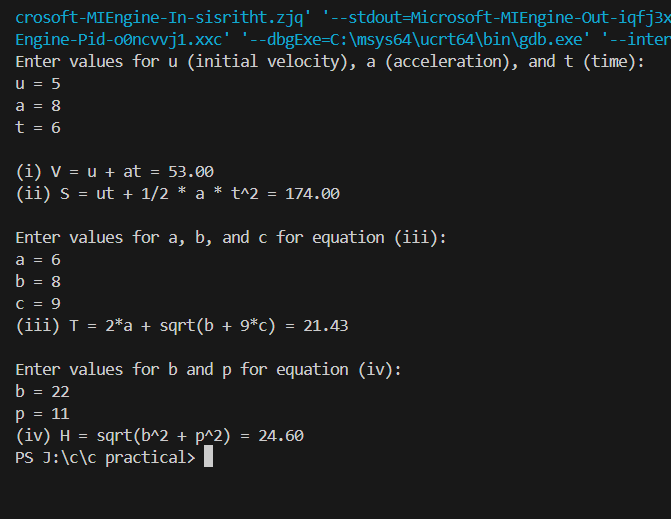
H = sqrt(b \* b + p \* p);

printf("(iv) H = sqrt(b^2 + p^2) = %.2f\n", H);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 7.**

**Aim:** Write a program to swap two variables:

a) By using temporary variable.

**Concept Used:** Variable declaration, assignment, and swapping using a temporary variable.

**Program**:

#include <stdio.h>

int main()

{

int a,b,temp;

printf("Before Swapping \n");

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

scanf("%d", &a);

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

scanf("%d", &b);

temp=a;

a=b;

b=temp;

printf("After Swapping\n");

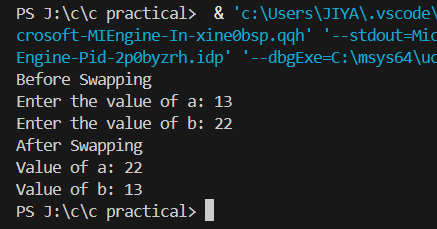
printf("Value of a: %d\n", a);

printf("Value of b: %d",b);

return 0;

}

**Output Screenshot:**



b) Without using temporary variable

**Concept Used:** Arithmetic swapping, bitwise XOR swapping.

**Program:**

#include <stdio.h>

int main() {

int a,b;

printf("Before Swapping\n");

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

scanf("%d", &a);

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

scanf("%d", &b);

a=a^b;

b=a^b;

a=a^b;

printf("After Swapping\n");

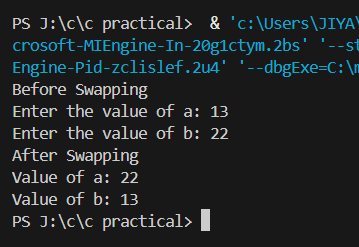
printf("Value of a: %d\n", a);

printf("Value of b: %d",b);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 8.**

**Aim:** Write a Program to find the greatest among three numbers using:

1. Conditional Operator

**Concept Used:**  Conditional operator (?:), comparisons.

**Program**:

a)

#include <stdio.h>

int main() {

int a, b, c, greatest;

printf("Enter three numbers: ");

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

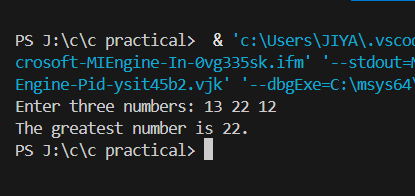
greatest = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);

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

return 0;

}

**Output Screenshot:**

****

1. If-Else statement

**Concept Used:** if-else statement, comparisons.

**Program:**

#include <stdio.h>

int main() {

int a,b,c;

printf("Enter first number a: ");

scanf("%d", &a);

printf("Enter second number b: ");

scanf("%d", &b);

printf("Enter third number c: ");

scanf("%d", &c);

if((a>b)&&(a>c))

{

printf("The greatest of three numbers is %d", a);

}

else if((b>a)&&(b>c))

{

printf("The greatest of three numbers is %d", b);

}

else{

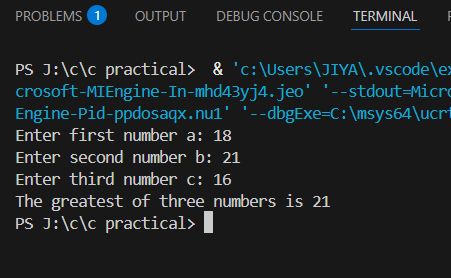
printf("The greatest of three numbers is %d", c);

}

return 0;

}

**Output Screenshot:**

****

**Experiment No. 9.**

**Aim:** Write the following programs using switch case statement:

1. To check that an input alphabet is vowel or consonant

**Concept Used:**  Switch-case structure, checking vowel/consonant, if-else statement.

**Program**:

#include <stdio.h>

#include <ctype.h>

int main()

{

char a,ch;

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

scanf("%c", &a);

ch= tolower(a);

switch (a){

case 'a': case 'e': case 'i' : case 'o': case'u':

printf("The alphabet is a vowel");

break;

default:

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

printf("The alphabet is a consonant");

}

else{

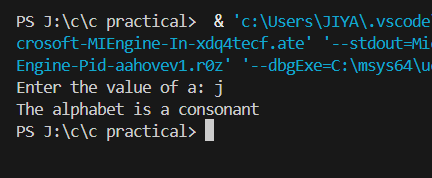
printf("Please enter a valid alphabet");

}

}

}

**Output Screenshot:**

****

1. To check whether a number is positive, negative or zero

**Concept Used:**  Switch-case structure, checking positive/negative/zero.

**Program:**

#include <stdio.h>

int main(){

int n;

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

scanf("%d", &n);

switch (n>0){

case 1:

printf("The number is positive");

break;

case 0:

switch(n<0){

case 1:

printf("The number is negative");

break;

case 0:

printf("The number is zero");

break;

}

default:

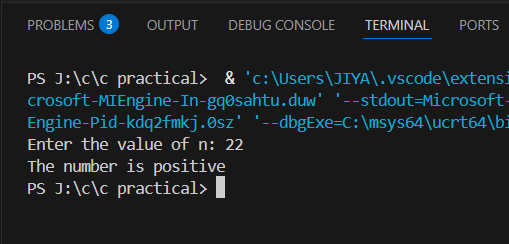
printf("Please enter a valid number");

}

return 0;

}

**Output Screenshot:**

****

**Experiment No. 10.**

**Aim:** Write a program using while loop to print the sum of first n natural numbers.

**Concept Used:** While loop, loop control, summation.

**Program**:

#include <stdio.h>

int main(){

int n, sum, i;

sum=0;

i=1;

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

scanf("%d", &n);

while(i<=n){

sum+=i;

i++;

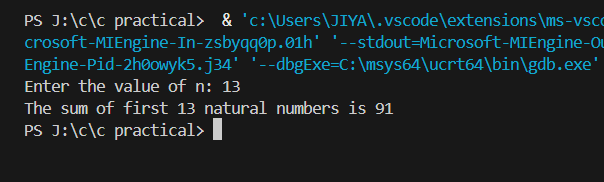
}

printf("The sum of first %d natural numbers is %d\n", n,sum);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 11.**

**Aim:** Write a program to check a number is Armstrong or not using For loop.

**Concept Used:** While loop, If-else statement , arithmetic operations, checking conditions, modular arithmetic.

**Program**:

#include <stdio.h>

#include <math.h>

int main(){

int a;

printf("Enter a number: ");

scanf("%d",&a);

int num,original;

int count=0;

num=a;

original=a;

while(num>0){

count++;

num=num/10;

}

int rem,armstrong=0;

while(a>0){

rem=a%10;

armstrong+=pow(rem,count);

a=a/10;

}

if(armstrong==original)

printf("The number is an armstrong number.");

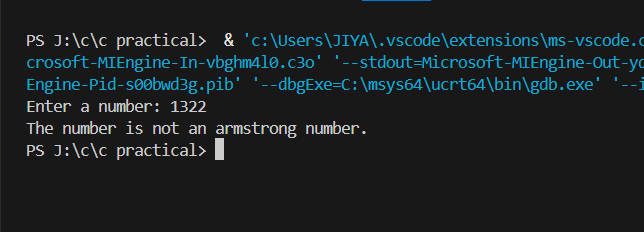
else

printf("The number is not an armstrong number.");

return 0;

}

**Output Screenshot:**

****

**Experiment No. 12.**

**Aim:** Write the program to count the digits in a number and then print the reverse of the number also.

**Concept Used:**  While loop, Loop control, arithmetic operations, modular arithmetic, reversing numbers.

**Program**:

#include <stdio.h>

int main() {

int num,originalNum,reversedNum=0, digitCount=0;

printf("Enter a number: ");

scanf("%d",&num);

originalNum=num;

while (num != 0) {

digitCount++;

reversedNum=reversedNum\*10 + num%10;

num/= 10;

}

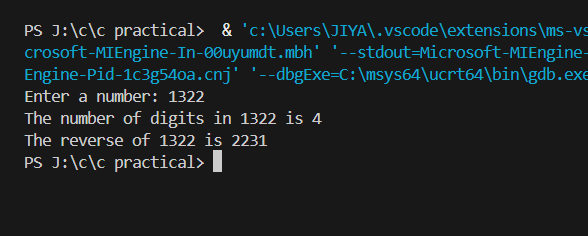
printf("The number of digits in %d is %d\n", originalNum, digitCount);

printf("The reverse of %d is %d\n", originalNum, reversedNum);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 13.**

**Aim:** Write a program to generate the Fibonacci series.

**Concept Used:** For Loop, sequence generation, variable assignments.

**Program**:

#include <stdio.h>

int main(){

int num;

printf("Enter a number: ");

scanf("%d",&num);

int a=0,b=1;

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

int temp=a;

printf("%d ",temp);

a=b;

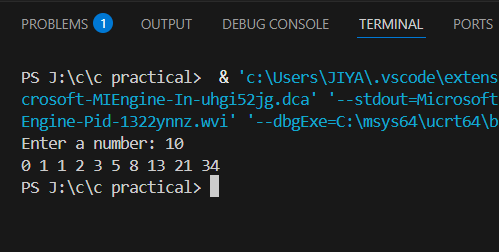
b=temp+a;

}

return 0;

}

**Output Screenshot:**

****

**Experiment No. 14.**

**Aim: Write a program to print the following patterns:**

**a)**

**\***

**\* \***

**\* \* \***

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

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

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

**Concept Used:** Nested loops, control structures, pattern printing.

**Program**:

#include <stdio.h>

int main() {

int n=6;

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

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

printf("\* ");

}

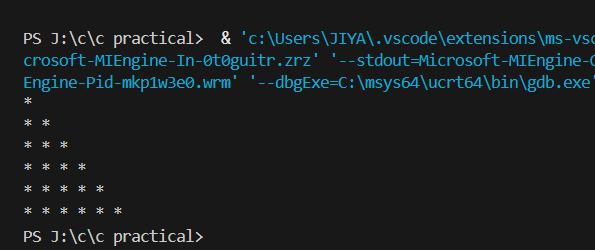
printf("\n");

}

return 0;

}

**Output Screenshot:**

****

**b)**

**\***

**\* \***

**\* \* \***

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

**\* \* \* \* \***

**\* \* \* \* \* \***

**Concept Used:** Nested loops, control structures, pattern printing.

**Program:**

#include <stdio.h>

int main() {

int n=6;

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

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

printf(" ");

}

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

printf("\* ");

}

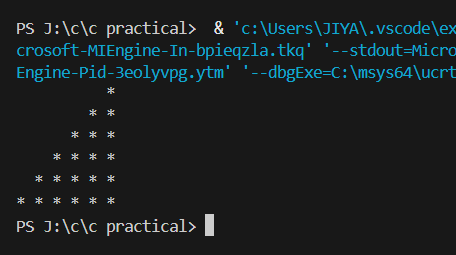
printf("\n");

}

return 0;

}

**Output Screenshot:**

****

**Experiment No. 15.**

**Aim:** Write the program to print the following pattern:

1  2    3     4    5     6

2  4    6     8    10   12

3  6    9    12   15   18

4  8    12  16   20   24

5  10  15  20   25   30

6  12  18  24   30   36

**Concept Used:** Nested loops, multiplication tables, formatting output.

**Program**:

#include <stdio.h>

int main() {

int n, i, j;

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

scanf("%d", &n);

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

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

printf(" %d", i \* j);

}

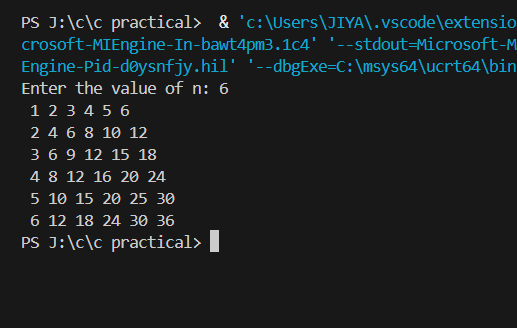
printf("\n");

}

return 0;

}

**Output Screenshot:**

****

**Experiment No. 16**

**Aim:** Write a program to check that the given number is prime, Armstrong or perfect using the concept of functions.

**Concept Used:** Functions, loops, conditional statements, and mathematical logic.

**Program**:

#include <stdio.h>

#include <math.h>

int isPrime(int n) {

if (n <= 1)

return 0;

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

if (n % i == 0)

return 0;

}

return 1;

}

int isArmstrong(int n) {

int original = n, sum = 0, digits = 0;

int temp = n;

while (temp > 0) {

digits++;

temp /= 10;

}

temp = n;

while (temp > 0) {

int digit = temp % 10;

sum += pow(digit, digits);

temp /= 10;

}

return sum == original;

}

int isPerfect(int n) {

int sum = 0;

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

if (n % i == 0)

sum += i;

}

return sum == n;

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (isPrime(num))

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

else

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

if (isArmstrong(num))

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

else

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

if (isPerfect(num))

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

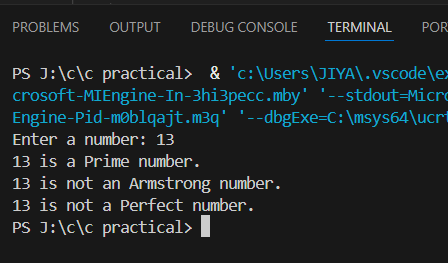
else

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

return 0;

}

**Output Screenshot:**

****

**Experiment No. 17**

**Aim:** Write a program to calculate the area and circumference of a circle using functions.

**Concept Used:** Functions, arithmetic operations, user input/output.

**Program**:

#include <stdio.h>

float calculateArea(float radius) {

return 3.1416 \* radius \* radius;

}

float calculateCircumference(float radius) {

return 2 \* 3.1416 \* radius;

}

int main() {

float r;

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

scanf("%f", &r);

float area = calculateArea(r);

float circumference = calculateCircumference(r);

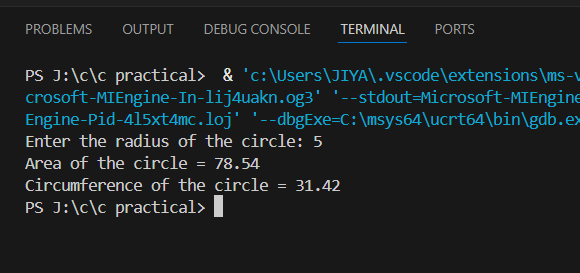
printf("Area of the circle = %.2f\n", area);

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

return 0;

}

**Output Screenshot:**

****

**Experiment No. 18**

**Aim:** Write a program to swap two variables using the concept of call by value and call by reference.

**Concept Used:** Call by value, call by reference (using pointers), functions, and user input/output

**Program**:

#include <stdio.h>

void swapByValue(int a, int b) {

int temp = a;

a = b;

b = temp;

printf("Inside swapByValue: a = %d, b = %d\n", a, b);

}

void swapByReference(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x, y;

printf("Enter two numbers: ");

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

printf("\nBefore swapByValue: x = %d, y = %d\n", x, y);

swapByValue(x, y);

printf("After swapByValue: x = %d, y = %d\n", x, y);

printf("\nBefore swapByReference: x = %d, y = %d\n", x, y);

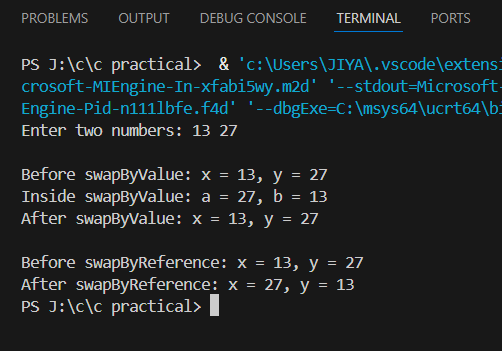
swapByReference(&x, &y);

printf("After swapByReference: x = %d, y = %d\n", x, y);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 19**

**Aim:** Write a program to perform the following operations on 1D-Array:

1. Insert
2. Update
3. Delete
4. Display
5. Search

**Concept Used:** Array manipulation, loops, user input/output, conditional statements, linear search

**Program**: #include <stdio.h>

void insert(int arr[], int \*n, int element, int position) {

for (int i = \*n; i >= position; i--) {

arr[i] = arr[i - 1];

}

arr[position] = element;

(\*n)++;

}

void update(int arr[], int n, int position, int newValue) {

if (position >= n || position < 0) {

printf("Invalid position!\n");

return;

}

arr[position] = newValue;

}

void delete(int arr[], int \*n, int position) {

if (position >= \*n || position < 0) {

printf("Invalid position!\n");

return;

}

for (int i = position; i < \*n - 1; i++) {

arr[i] = arr[i + 1];

}

(\*n)--;

}

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

if (n == 0) {

printf("Array is empty!\n");

return;

}

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

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

}

}

void search(int arr[], int n, int element) {

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

if (arr[i] == element) {

printf("Element found at position %d.\n", i);

return;

}

}

printf("Element not found.\n");

}

int main() {

int arr[100] = {10, 20, 30, 40, 50};

int n = 5;

int choice, element, position, newValue;

printf("Initial Array: ");

display(arr, n);

printf("\n");

while (1) {

printf("\n1. Insert\n2. Update\n3. Delete\n4. Display\n5. Search\n6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter element to insert: ");

scanf("%d", &element);

printf("Enter position to insert (0 to %d): ", n);

scanf("%d", &position);

insert(arr, &n, element, position);

printf("Updated Array: ");

display(arr, n);

printf("\n");

break;

case 2:

printf("Enter position to update: ");

scanf("%d", &position);

printf("Enter new value: ");

scanf("%d", &newValue);

update(arr, n, position, newValue);

printf("Updated Array: ");

display(arr, n);

printf("\n");

break;

case 3:

printf("Enter position to delete: ");

scanf("%d", &position);

delete(arr, &n, position);

printf("Updated Array: ");

display(arr, n);

printf("\n");

break;

case 4:

printf("Current Array: ");

display(arr, n);

printf("\n");

break;

case 5:

printf("Enter element to search: ");

scanf("%d", &element);

search(arr, n, element);

break;

case 6:

return 0;

default:

printf("Invalid choice!\n");

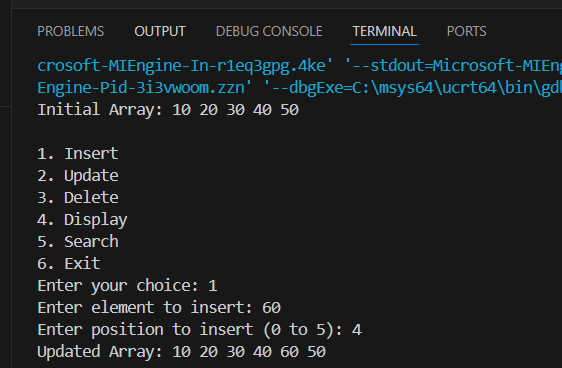
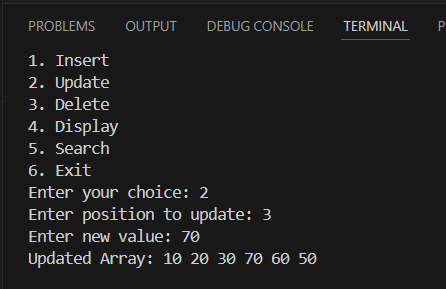
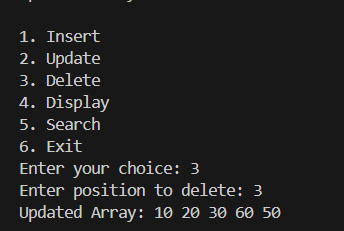
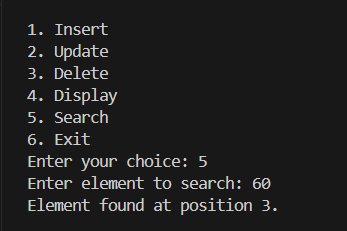
}

}

return 0;

}

**Output Screenshot:**

**** ****   

**Experiment No. 20.**

**Aim:** Write a program to calculate the sum of array elements by passing it to a function.

**Concept Used:** Function, arrays, looping, passing arrays to functions, return values.

**Program**:

#include <stdio.h>

int sum\_of\_elements(int arr[], int n) {

int sum = 0;

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

sum += arr[i];

}

return sum;

}

int main() {

int arr[] = {10, 20, 30, 40, 50};

int n = sizeof(arr) / sizeof(arr[0]);

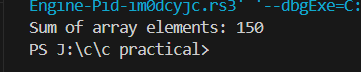
int result = sum\_of\_elements(arr, n);

printf("Sum of array elements: %d\n", result);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 21.**

**Aim:** Write a program to show the use of passing pointer as arguments to the functions.

**Concept Used:** Pointers, function arguments, pass-by-reference, memory address manipulation, swapping values using pointers.

**Program**:

#include <stdio.h>

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x = 10, y = 20;

printf("Before swap: x = %d, y = %d\n", x, y);

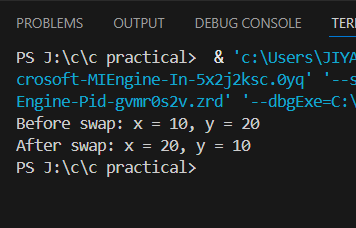
swap(&x, &y);

printf("After swap: x = %d, y = %d\n", x, y);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 22.**

**Aim:** Write a program matrix multiplication using the concept of 2D array

**Concept Used:** 2D arrays, matrix multiplication, loops, user input/output, conditional statements.

**Program**:

#include <stdio.h>

void multiply\_matrices(int first[][100], int second[][100], int result[][100], int first\_rows, int first\_cols, int second\_cols) {

for (int i = 0; i < first\_rows; i++) {

for (int j = 0; j < second\_cols; j++) {

result[i][j] = 0;

for (int k = 0; k < first\_cols; k++) {

result[i][j] += first[i][k] \* second[k][j];

}

}

}

}

void display\_matrix(int matrix[][100], int rows, int cols) {

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

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

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

}

printf("\n");

}

}

int main() {

int first\_rows, first\_cols, second\_rows, second\_cols;

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

scanf("%d %d", &first\_rows, &first\_cols);

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

scanf("%d %d", &second\_rows, &second\_cols);

if (first\_cols != second\_rows) {

printf("Matrix multiplication is not possible. Columns of first matrix must be equal to rows of second matrix.\n");

return 0;

}

int first[100][100], second[100][100], result[100][100];

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

for (int i = 0; i < first\_rows; i++) {

for (int j = 0; j < first\_cols; j++) {

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

}

}

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

for (int i = 0; i < second\_rows; i++) {

for (int j = 0; j < second\_cols; j++) {

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

}

}

multiply\_matrices(first, second, result, first\_rows, first\_cols, second\_cols);

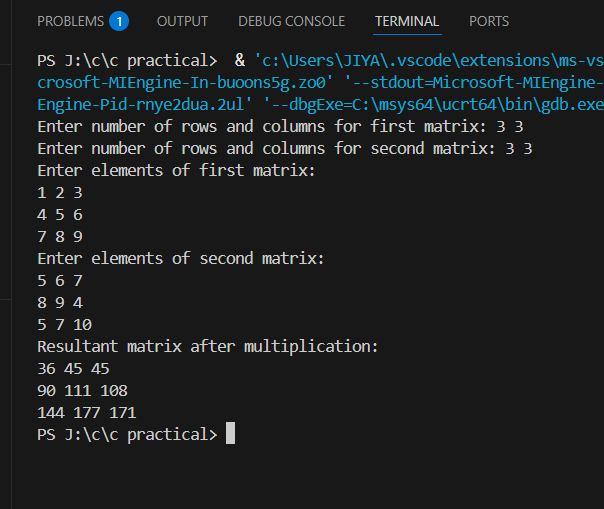
printf("Resultant matrix after multiplication:\n");

display\_matrix(result, first\_rows, second\_cols);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 23**

**Aim:** Write a program to transpose a given matrix.

**Concept Used:** 2D arrays, matrix transpose, loops, user input/output.

**Program**:

#include <stdio.h>

void transpose\_matrix(int matrix[][10], int transposed[][10], int rows, int cols) {

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

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

transposed[j][i] = matrix[i][j];

}

}

}

void display\_matrix(int matrix[][10], int rows, int cols) {

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

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

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

}

printf("\n");

}

}

int main() {

int matrix[10][10], transposed[10][10];

int rows, cols;

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

scanf("%d %d", &rows, &cols);

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

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

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

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

}

}

transpose\_matrix(matrix, transposed, rows, cols);

printf("Original Matrix:\n");

display\_matrix(matrix, rows, cols);

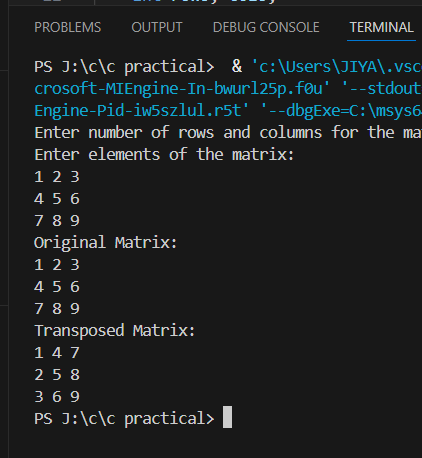
printf("Transposed Matrix:\n");

display\_matrix(transposed, cols, rows);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 24.**

**Aim:** Write a program to find the factorial of a number by using the concept of recursion.

**Concept Used:** Recursion, base case, recursive function calls

**Program**:

#include <stdio.h>

int factorial(int n) {

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

return 1;

}

return n \* factorial(n - 1);

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (num < 0) {

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

} else {

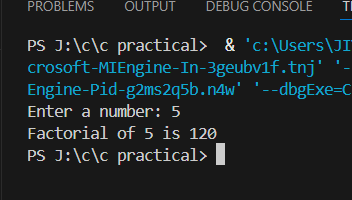
printf("Factorial of %d is %d\n", num, factorial(num));

}

return 0;

}

**Output Screenshot:**

****

**Experiment No. 25.**

**Aim:** Write a menu driven C program to show the use of in-built string functions like strlen, strcat, strcpy, strcmp, strrev etc.

**Concept Used:** Menu-driven program structure, in-built string functions (strlen, strcat, strcpy, strcmp, strrev), string manipulation, fgets() for safe input, getchar() for handling newline characters, conditional statements, loops.

**Program**:

#include <stdio.h>

#include <string.h>

int main() {

    char str1[100] = "Jiya", str2[100] = "Manchanda", result[200];

    printf("Original String 1: %s\n", str1);

    printf("Original String 2: %s\n", str2);

    // strlen

    printf("\nLength of String 1: %zu\n", strlen(str1));

    // strcpy

    strcpy(result, str1);

    printf("\nCopy of String 1 into result: %s\n", result);

    // strcat

    strcat(result, str2);

    printf("\nConcatenated String (result + str2): %s\n", result);

    // strcmp

    int cmp = strcmp(str1, str2);

    if (cmp == 0) {

        printf("\nStrings are equal.\n");

    } else if (cmp < 0) {

        printf("\nString 1 is less than String 2.\n");

    } else {

        printf("\nString 1 is greater than String 2.\n");

    }

    // strrev (note: this is non-standard, may not work in all compilers)

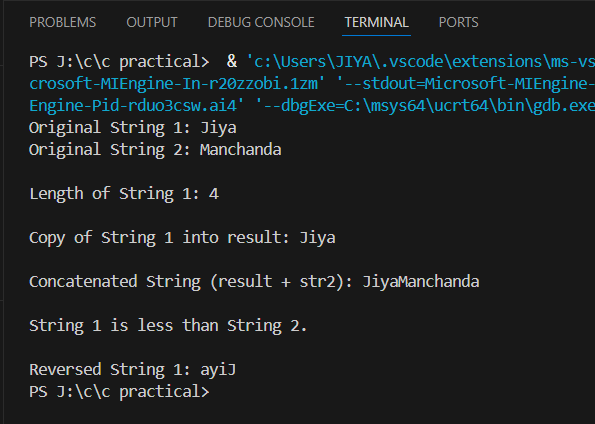
    strrev(str1);

    printf("\nReversed String 1: %s\n", str1);

    return 0;

}

**Output Screenshot:**

****

**Experiment No. 26.**

**Aim:** Write a Program in C to display the total number of appearances of a substring provided as input by the user in a given string.

**Concept Used:** String manipulation, substring search using strncmp, loop traversal, string input using fgets, string length calculation using strlen.

**Program**:

#include <stdio.h>

#include <string.h>

int main() {

    char str[200] = "JiyaManchanda";

    char sub[50];

    int count = 0;

    int len, sub\_len;

    printf("Main string: %s\n", str);

    printf("Enter substring to search: ");

    fgets(sub, sizeof(sub), stdin);

    sub[strcspn(sub, "\n")] = '\0';

    len = strlen(str);

    sub\_len = strlen(sub);

    for (int i = 0; i <= len - sub\_len; i++) {

        if (strncmp(&str[i], sub, sub\_len) == 0) {

            count++;

        }

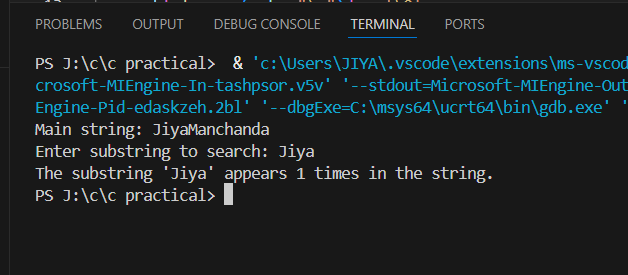
    }

    printf("The substring '%s' appears %d times in the string.\n", sub, count);

    return 0;

}

**Output Screenshot:**

****

**Experiment No. 27.**

**Aim:** Write a program to display the sum of the digits of a number by using the concept of recursion.

**Concept Used:** Recursion, digit extraction using modulus and division, user input, base case in recursion

**Program**:

#include <stdio.h>

int sum\_of\_digits(int n) {

if (n == 0)

return 0;

return (n % 10) + sum\_of\_digits(n / 10);

}

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (num < 0)

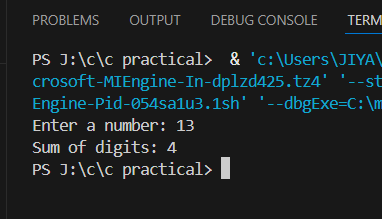
num = -num; // Handle negative numbers

printf("Sum of digits: %d\n", sum\_of\_digits(num));

return 0;

}

**Output Screenshot:**

****

**Experiment No. 28.**

**Aim:** Write a C program to add two distances in inch & feet using the concept of structures.

**Concept Used:** Structures, structure variables, structure arithmetic, inches-to-feet conversion, initialization of structure data.

**Program**:

#include <stdio.h>

struct Distance {

int feet;

int inches;

};

int main() {

struct Distance d1 = {5, 8};

struct Distance d2 = {6, 11};

struct Distance result;

result.feet = d1.feet + d2.feet;

result.inches = d1.inches + d2.inches;

if (result.inches >= 12) {

result.feet += result.inches / 12;

result.inches = result.inches % 12;

}

printf("First distance: %d feet %d inches\n", d1.feet, d1.inches);

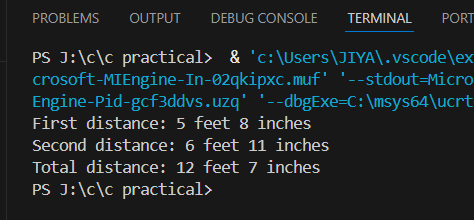
printf("Second distance: %d feet %d inches\n", d2.feet, d2.inches);

printf("Total distance: %d feet %d inches\n", result.feet, result.inches);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 29.**

**Aim:** Write a C program to add two complex numbers using the concept of structures in C.

**Concept Used:** Structures, structure variables, arithmetic on structure fields, complex number addition.

**Program**:

#include <stdio.h>

struct Complex {

float real;

float imag;

};

int main() {

struct Complex c1 = {3.2, 4.5};

struct Complex c2 = {1.8, 2.5};

struct Complex sum;

sum.real = c1.real + c2.real;

sum.imag = c1.imag + c2.imag;

printf("First complex number: %.1f + %.1fi\n", c1.real, c1.imag);

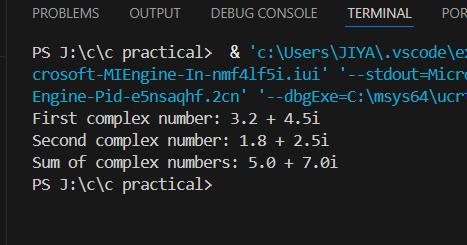
printf("Second complex number: %.1f + %.1fi\n", c2.real, c2.imag);

printf("Sum of complex numbers: %.1f + %.1fi\n", sum.real, sum.imag);

return 0;

}

**Output Screenshot:**

****

**Experiment No. 30.**

**Aim:** Write a program in C to store the information of five employees using both concepts i.e. array of structure and array within structure

**Concept Used:** Array of structures, array within structure, structured data storage, fixed data initialization, nested loops for structured printing.

**Program**:

#include <stdio.h>

struct Employee {

int id;

char name[50];

int phone[10];

};

int main() {

struct Employee emp[5] = {

{101, "Alice", {9,8,7,6,5,4,3,2,1,0}},

{102, "Bob", {9,9,8,8,7,7,6,6,5,5}},

{103, "Charlie", {1,2,3,4,5,6,7,8,9,0}},

{104, "Daisy", {2,2,2,2,2,2,2,2,2,2}},

{105, "Ethan", {0,1,2,3,4,5,6,7,8,9}}

};

printf("Employee Details:\n");

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

printf("\nID: %d\n", emp[i].id);

printf("Name: %s\n", emp[i].name);

printf("Phone: ");

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

printf("%d", emp[i].phone[j]);

}

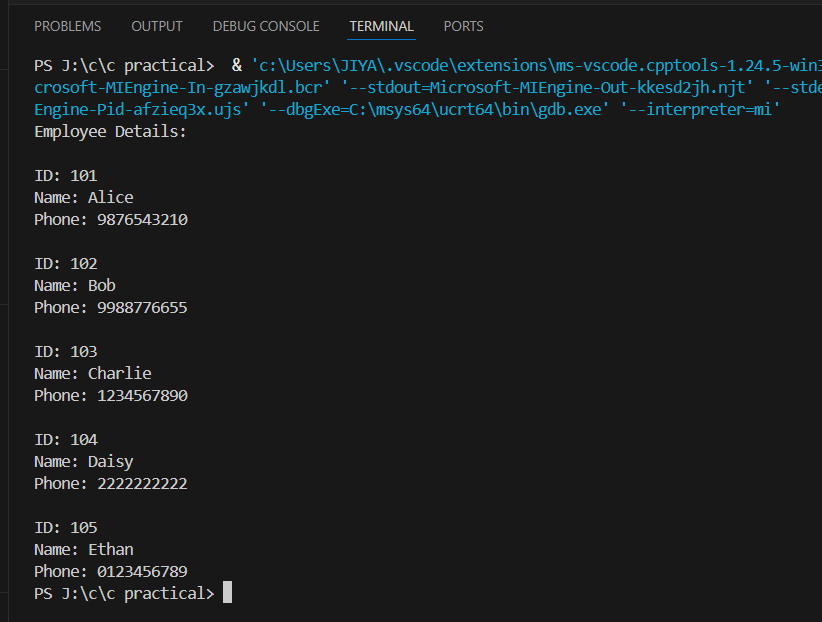
printf("\n");

}

return 0;

}

**Output Screenshot:**

****

**Experiment No. 31.**

**Aim:** Write a Program in C to find and replace a specific string in a file and also display the total number of appearances of that string.

**Concept Used:**

File handling, string comparison and replacement, reading/writing files, word-by-word processing, use of fscanf, fprintf, and strcmp.

**Program**:

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main() {

FILE \*fp1, \*fp2;

char ch, word[100], temp[1000];

char find[100] = "old", replace[100] = "new";

int count = 0;

fp1 = fopen("input.txt", "r");

fp2 = fopen("output.txt", "w");

if (fp1 == NULL || fp2 == NULL) {

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

return 1;

}

while (fscanf(fp1, "%s", word) != EOF) {

if (strcmp(word, find) == 0) {

fprintf(fp2, "%s ", replace);

count++;

} else {

fprintf(fp2, "%s ", word);

}

}

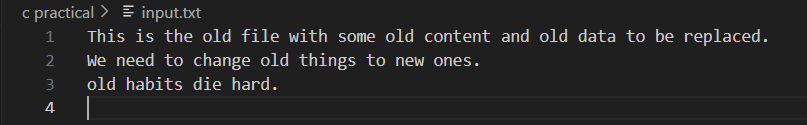
fclose(fp1);

fclose(fp2);

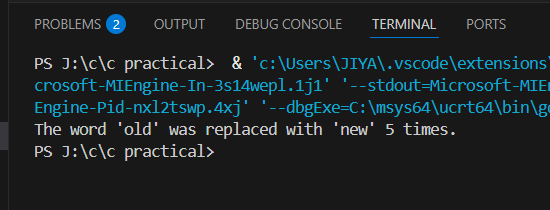
printf("The word '%s' was replaced with '%s' %d times.\n", find, replace, count);

return 0;

}



**Output Screenshot:**

****