MODULE 3:

1. Write a Program to check the given year is leap year or not.

#include <stdio.h>

int main() {

int year;

// Taking user input for the year

printf("Enter a year: ");

scanf("%d", &year);

// Checking if it's a leap year

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

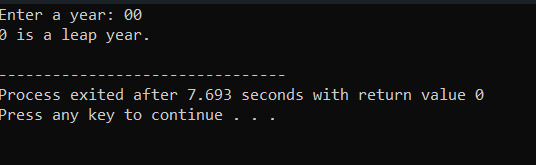
printf("%d is a leap year.\n", year);

} else {

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

}

return 0;}



1. **Write a Program to check the given number is prime or not prime.**

#include <stdio.h>

int isPrime(int num) {

if (num <= 1) {

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

}

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

if (num % i == 0) {

return 0; // Not a prime number

}

}

return 1; // Prime number

}

int main() {

int number;

// Taking user input

printf("Enter a number: ");

scanf("%d", &number);

if (isPrime(number)) {

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

} else {

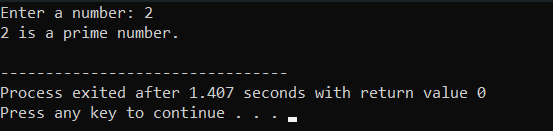
printf("%d is not a prime number.\n", number);

}

return 0;

}

Out put:



**3. Write a program user enter the 5 subjects mark. You have to make a total and find the percentage. percentage > 75 you have to print “Distinction” percentage > 60 and percentage <= 75 you have to print “First class” percentage >50 and percentage <= 60 you have to print “Second class” percentage > 35 and percentage <= 50 you have to print “Pass class” Otherwise print “Fail”**

#include <stdio.h>

int main() {

int marks[5];

int total = 0;

float percentage;

// Taking input for 5 subjects' marks

printf("Enter the marks for 5 subjects:\n");

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

printf("Enter marks for subject %d: ", i + 1);

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

total += marks[i];

}

// Calculating percentage

percentage = (float)total / 5;

// Printing the percentage

printf("Total marks: %d\n", total);

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

// Checking the percentage and determining the grade

if (percentage > 75) {

printf("Grade: Distinction\n");

} else if (percentage > 60 && percentage <= 75) {

printf("Grade: First class\n");

} else if (percentage > 50 && percentage <= 60) {

printf("Grade: Second class\n");

} else if (percentage > 35 && percentage <= 50) {

printf("Grade: Pass class\n");

} else {

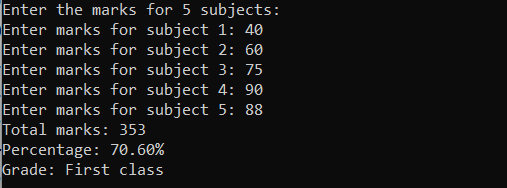
printf("Grade: Fail\n");

}

return 0

}

**Out put:**



**4. Write a program of to find out the Area of Triangle, Rectangle and Circle using Switch Case .(Must Be Menu Driven)**

#include <stdio.h>

#define PI 3.14159

int main() {

int choice;

float base, height, width, length, radius, area;

do {

printf("\nMenu:\n");

printf("1. Area of Triangle\n");

printf("2. Area of Rectangle\n");

printf("3. Area of Circle\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice

switch (choice) {

case 1:

printf("Enter base of the triangle: ");

scanf("%f", &base);

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

scanf("%f", &height);

area = 0.5 \* base \* height;

printf("Area of the triangle is: %.2f\n", area);

break;

case 2:

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

scanf("%f", &width);

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

scanf("%f", &length);

area = width \* length;

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

break;

case 3:

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

scanf("%f", &radius);

area = PI \* radius \* radius;

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

break;

case 4:

printf("Exiting the program. Goodbye!\n");

break;

default:

printf("Invalid choice! Please enter a valid option.\n");

break;

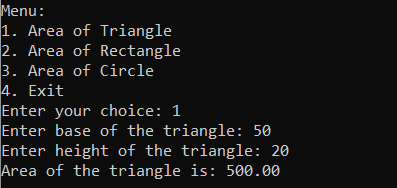
}

} while (choice != 4);

return 0;

}

Out put:



**5. Write a program you have to print the Fibonacci series up to user given number**

#include <stdio.h>

void printFibonacci(int n) {

int first = 0, second = 1, next;

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

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

printf("%d, ", first);

next = first + second;

first = second;

second = next;

}

printf("\n");

}

int main() {

int terms;

printf("Enter the number of terms for the Fibonacci series: ");

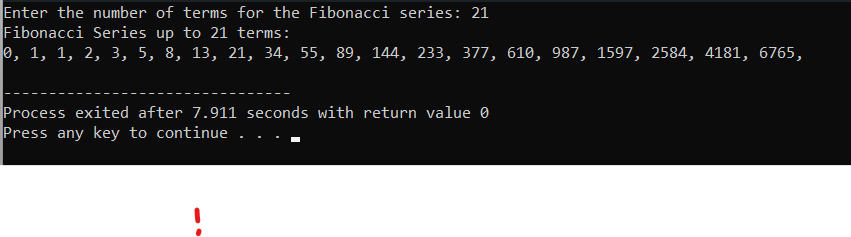
scanf("%d", &terms);

printFibonacci(terms);

return 0;

}

Out put:



6. Write a program to print the number in reverse order.

#include <stdio.h>

int main() {

int num, reversedNum = 0, remainder;

// Input

printf("Enter an integer: ");

scanf("%d", &num);

// Reversing the number

while (num != 0) {

remainder = num % 10;

reversedNum = reversedNum \* 10 + remainder;

num /= 10;

}

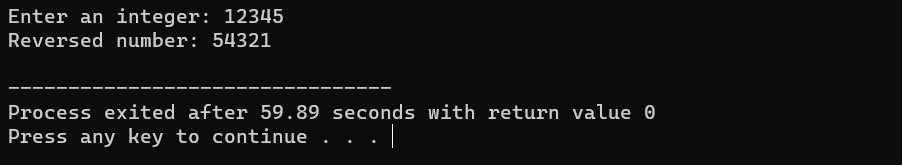
//

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

return 0;

}

Output:



7. Write a program make a summation of given number(E.g. 1523 ans :-11)

#include <stdio.h>

int main() {

int num, digit, sum = 0;

// Input

printf("Enter an integer: ");

scanf("%d", &num);

// Calculating the sum of digits

while (num != 0) {

digit = num % 10;

sum += digit;

num /= 10;

}

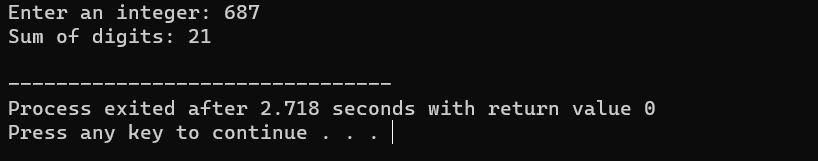
// Output

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

return 0;

}

Out put:



8. Half Pyramid

**A}**

#include <stdio.h>

int main() {

int rows, i, j;

// Input

printf("Enter number of rows: ");

scanf("%d", &rows);

// Printing half pyramid

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

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

printf("\* ")

}

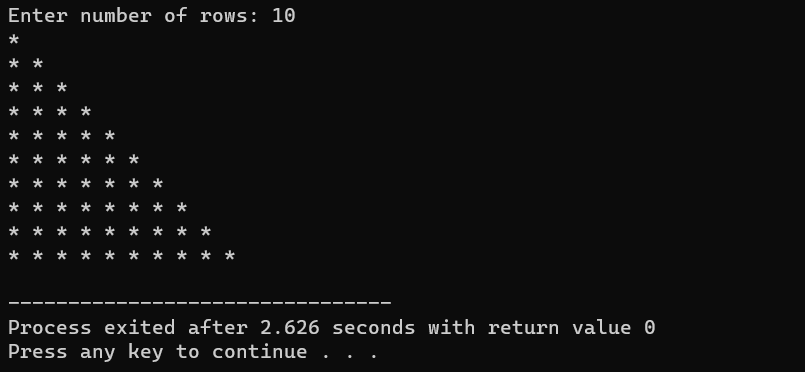
printf("\n");

}

return 0;

}

Out put:



**B} Inverted full pyramid of \***

**#include <stdio.h>**

**int main() {**

**int rows, i, j, space;**

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

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

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

**for (space = 0; space < rows - i; ++space)**

**printf(" ");**

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

**printf("\* ");**

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

**printf("\* ");**

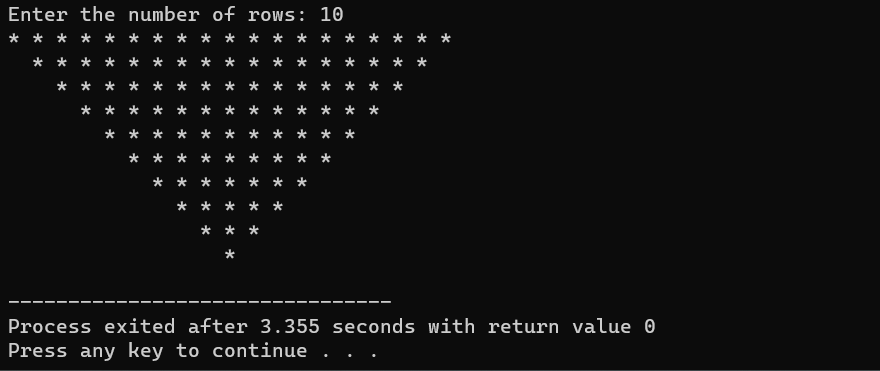
**printf("\n");**

**}**

**return 0;**

**}**

**Out put:**



**C}**  Full Pyramid of Numbers

#include <stdio.h>

int main() {

int i, space, rows, k = 0, count = 0, count1 = 0;

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

scanf("%d", &rows);

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

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

printf(" ");

++count;

}

while (k != 2 \* i - 1) {

if (count <= rows - 1) {

printf("%d ", i + k);

++count;

} else {

++count1;

printf("%d ", (i + k - 2 \* count1));

}

++k;

}

count1 = count = k = 0;

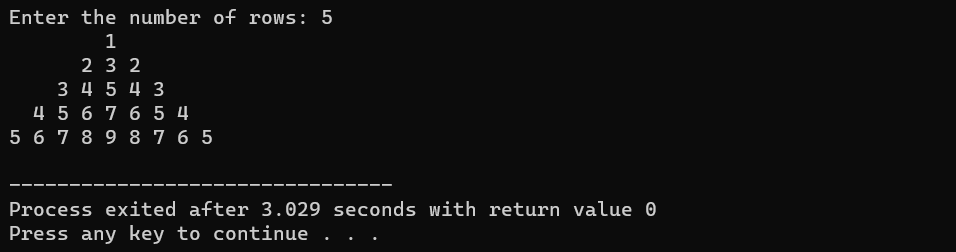
printf("\n");

}

return 0;

}

Out put:

****

**9.** Write a program to find out the max number from given 10 elements of array

#include <stdio.h>

int main() {

int arr[10];

int i, max;

// Input 10 elements

printf("Enter 10 elements:\n");

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

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

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

}

// Assume the first element as the maximum

max = arr[0];

// Iterate through the array to find the maximum

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

if (arr[i] > max) {

max = arr[i];

}

}

// Display the maximum number

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

return 0;

}

**Out put:**

****

**10.** **Write a program to sort the array of 5 elements.**

#include <stdio.h>

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

int i, j, temp;

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

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

if (arr[j] > arr[j + 1]) {

// Swap arr[j] and arr[j + 1]

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int main() {

int arr[5];

int i;

// Input 5 elements

printf("Enter 5 elements:\n");

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

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

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

}

// Sort the array

bubbleSort(arr, 5);

// Display the sorted array

printf("Sorted array in ascending order: ");

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

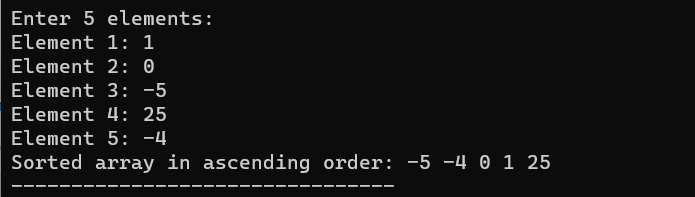
printf("%d ", arr[i]);

}

return 0;

}

Out put:



11**. Write a Program of find the element of given position from the array**

#include <stdio.h>

int main() {

int arr[50], size, position;

// Input the size of the array

printf("Enter the size of the array (not more than 50): ");

scanf("%d", &size);

if (size <= 0 || size > 50) {

printf("Invalid array size. Please enter a size between 1 and 50.\n");

return 1; // Exit the program with an error code

}

// Input array elements

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

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

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

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

}

// Input the position of the element to find

printf("Enter the position of the element to find (1 to %d): ", size);

scanf("%d", &position);

// Check if the position is valid

if (position < 1 || position > size) {

printf("Invalid position. Please enter a position between 1 and %d.\n", size);

return 1; // Exit the program with an error code

}

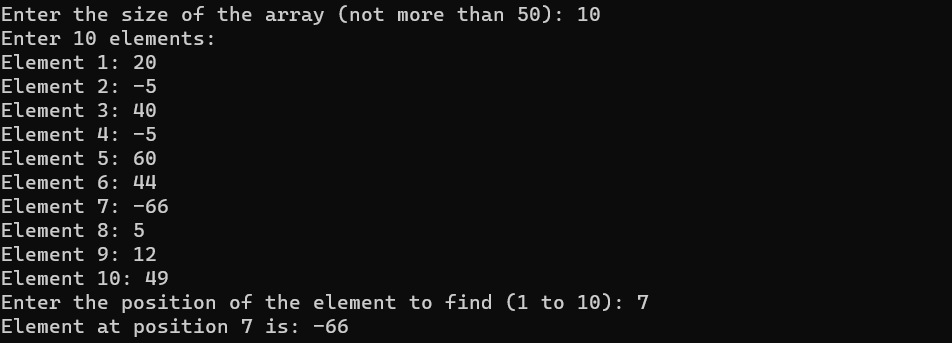
// Find and display the element at the given position

printf("Element at position %d is: %d\n", position, arr[position - 1]);

return 0;

}

Out put:



**12.** **Write a program to find out the Max number from given Matrix**

#include <stdio.h>

#define MAX\_ROWS 3

#define MAX\_COLS 3

int main() {

int matrix[MAX\_ROWS][MAX\_COLS];

int max, i, j;

// Input the matrix elements

printf("Enter the matrix elements (%d x %d):\n", MAX\_ROWS, MAX\_COLS);

for (i = 0; i < MAX\_ROWS; i++) {

for (j = 0; j < MAX\_COLS; j++) {

printf("Element at position (%d, %d): ", i + 1, j + 1);

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

}

}

// Assume the first element as the maximum

max = matrix[0][0];

// Iterate through the matrix to find the maximum

for (i = 0; i < MAX\_ROWS; i++) {

for (j = 0; j < MAX\_COLS; j++) {

if (matrix[i][j] > max) {

max = matrix[i][j];

}

}

}

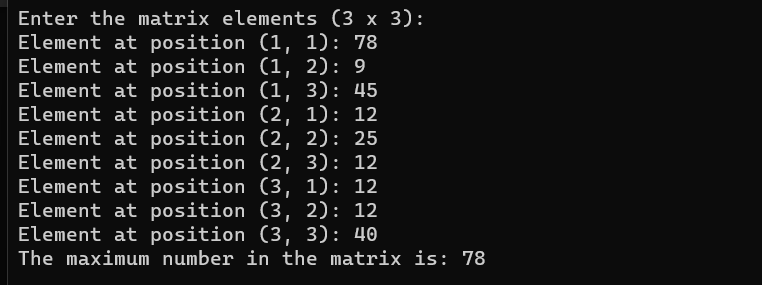
// Display the maximum number

printf("The maximum number in the matrix is: %d\n", max);

return 0

}

Out put:



13. **Write a program to copy string from one string to another string with user define function**

**#include <stdio.h>**

**// Function to copy a string from source to destination**

**void stringCopy(char \*destination, const char \*source) {**

**// Iterate through each character and copy to destination**

**while (\*source != '\0') {**

**\*destination = \*source;**

**source++;**

**destination++;**

**}**

**// Add null terminator to the destination string**

**\*destination = '\0';**

**}**

**int main() {**

**char sourceString[100], destinationString[100];**

**// Input the source string**

**printf("Enter the source string: ");**

**scanf("%s", sourceString);**

**// Call the user-defined function to copy the string**

**stringCopy(destinationString, sourceString);**

**// Display the original and copied strings**

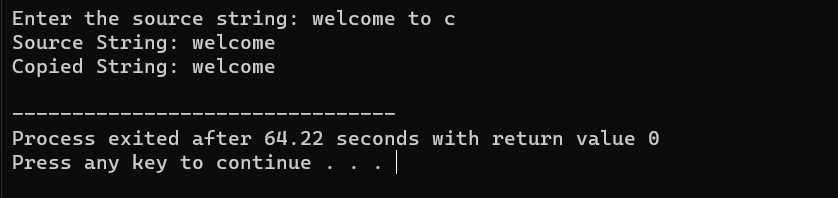
**printf("Source String: %s\n", sourceString);**

**printf("Copied String: %s\n", destinationString);**

**return 0;**

**}**

**Out put:**

****

**14**. Write a program to find out the factorial of given number using function.

**#include <stdio.h>**

**// Function to calculate the factorial of a number**

**unsigned long long factorial(int num) {**

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

**return 1; // Base case: 0! and 1! are both 1**

**} else {**

**// Recursive case: num! = num \* (num-1)!**

**return num \* factorial(num - 1);**

**}**

**}**

**int main() {**

**int number;**

**// Input the number for which factorial needs to be calculated**

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

**scanf("%d", &number);**

**// Check if the number is non-negative**

**if (number < 0) {**

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

**} else {**

**// Call the function to calculate factorial**

**unsigned long long result = factorial(number);**

**// Display the result**

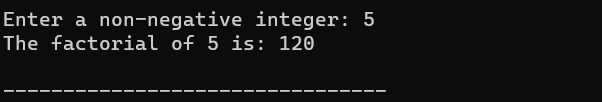
**printf("The factorial of %d is: %llu\n", number, result);**

**}**

**return 0;**

**}**

**Out put:**

****

**15. Write a program to print the Fibonacci series using function**

#include <stdio.h>

// Function to print Fibonacci series up to n terms

void printFibonacci(int n) {

int first = 0, second = 1, next;

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

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

printf("%d, ", first);

next = first + second;

first = second;

second = next;

}

printf("\n");

}

int main() {

int terms;

// Input the number of terms for Fibonacci series

printf("Enter the number of terms for Fibonacci series: ");

scanf("%d", &terms);

// Check if the number of terms is valid

if (terms <= 0) {

printf("Number of terms should be a positive integer.\n");

} else {

// Call the function to print Fibonacci series

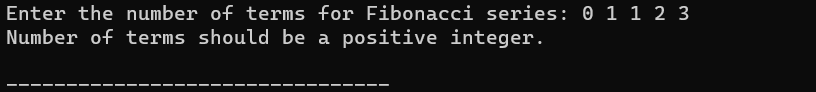
printFibonacci(terms);

}

return 0;

}

Out put:



16**. Write a Program of Print a number and check the number is palindrome or not using recursive function.**

#include <stdio.h>

#include <conio.h>

int reverse(int num);

int isPalindrome(int num);

int main() {

int num;

printf("Enter a number to check Palindrome: ");

scanf("%d", & num);

if (isPalindrome(num) == 1) {

printf("The given number is a Palindrome");

} else {

printf("The given number is not a Palindrome");

}

return 0;

}

int isPalindrome(int num) {

if (num == reverse(num)) {

return 1;

}

return 0;

}

int reverse(int num) {

int rem;

static int sum = 0;

if (num != 0) {

rem = num % 10;

sum = sum \* 10 + rem;

reverse(num / 10);

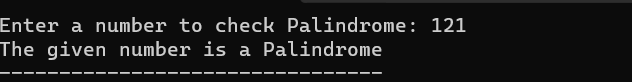
} else

return sum;

return sum;

}

Out put:



17. **Write a program of structure for five employee that provides the following information print and display empno, empname, address and age**

#include <stdio.h>

// Define a structure for employee information

struct Employee {

int empno;

char empname[50];

char address[100];

int age;

};

int main() {

// Declare an array of structures to store information for five employees

struct Employee employees[5];

// Input information for each employee

printf("Enter information for five employees:\n");

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

printf("\nEmployee %d:\n", i + 1);

printf("Enter Employee Number: ");

scanf("%d", &employees[i].empno);

printf("Enter Employee Name: ");

scanf("%s", employees[i].empname);

printf("Enter Address: ");

scanf("%s", employees[i].address);

printf("Enter Age: ");

scanf("%d", &employees[i].age);

}

// Display information for each employee

printf("\nEmployee Information:\n");

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

printf("\nEmployee %d:\n", i + 1);

printf("Employee Number: %d\n", employees[i].empno);

printf("Employee Name: %s\n", employees[i].empname);

printf("Address: %s\n", employees[i].address);

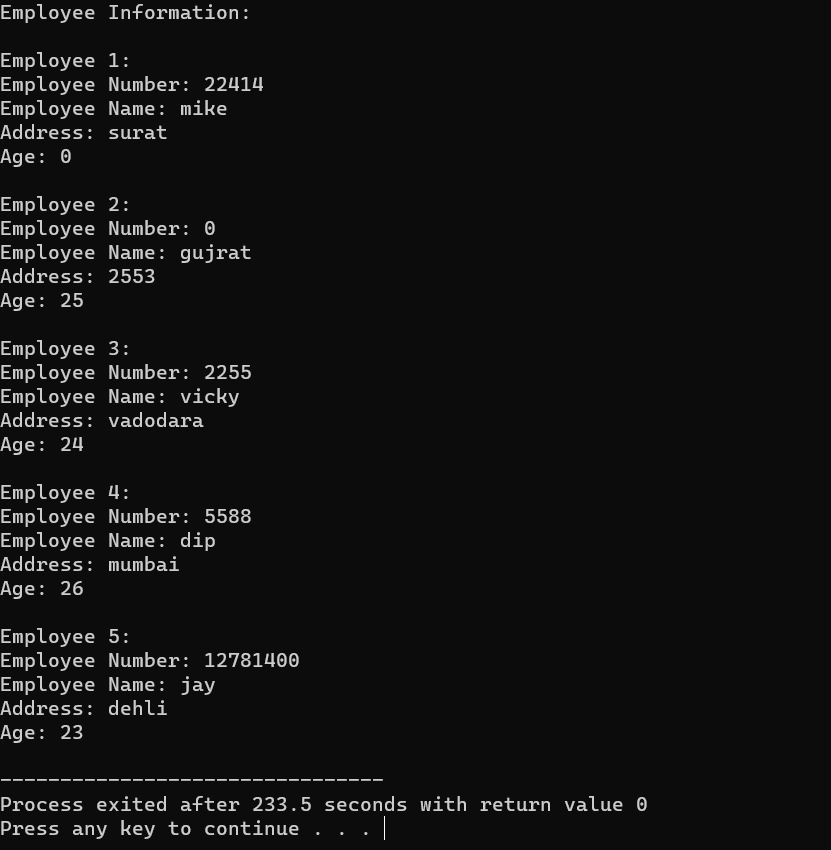
printf("Age: %d\n", employees[i].age);

}

return 0;

}

Out put:



**18. Write program to make a addition of two number using pointer.**

#include <stdio.h>

// Function to add two numbers using pointers

void addNumbers(int \*a, int \*b, int \*sum) {

\*sum = \*a + \*b;

}

int main() {

int num1, num2, result;

// Input two numbers

printf("Enter the first number: ");

scanf("%d", &num1);

printf("Enter the second number: ");

scanf("%d", &num2);

// Call the function to add numbers using pointers

addNumbers(&num1, &num2, &result);

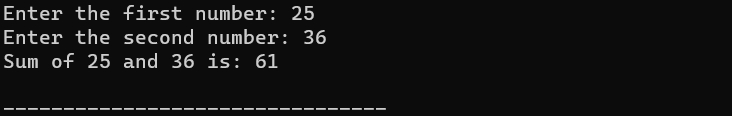
// Display the result

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

return 0;

}

**Out put:**

****

19**. Write a program to concatenate the two string using pointer.**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Function to concatenate two strings using pointers

char \*concatenateStrings(const char \*str1, const char \*str2) {

// Calculate the length of the concatenated string

int len1 = strlen(str1);

int len2 = strlen(str2);

int totalLength = len1 + len2;

// Allocate memory for the concatenated string

char \*concatenatedString = (char \*)malloc((totalLength + 1) \* sizeof(char));

// Check if memory allocation was successful

if (concatenatedString == NULL) {

printf("Memory allocation failed.\n");

exit(1);

}

// Copy the first string to the concatenated string

strcpy(concatenatedString, str1);

// Concatenate the second string to the concatenated string

strcat(concatenatedString, str2);

return concatenatedString;

}

int main() {

char str1[50], str2[50];

// Input two strings

printf("Enter the first string: ");

scanf("%s", str1);

printf("Enter the second string: ");

scanf("%s", str2);

// Call the function to concatenate strings using pointers

char \*result = concatenateStrings(str1, str2);

// Display the concatenated string

printf("Concatenated String: %s\n", result);

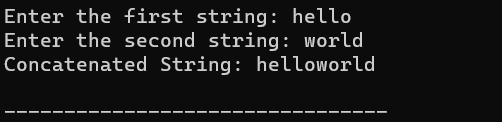
// Free the memory allocated for the concatenated string

free(result);

return 0;

}

Out put:



20. . **Write a program to sort the numbers using pointer and functions**

#include <stdio.h>

// Function to perform Bubble Sort on an array using pointers

void bubbleSort(int \*arr, int size) {

int i, j, temp;

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

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

if (\*(arr + j) > \*(arr + j + 1)) {

// Swap arr[j] and arr[j + 1]

temp = \*(arr + j);

\*(arr + j) = \*(arr + j + 1);

\*(arr + j + 1) = temp;

}

}

}

}

int main() {

int size, i;

// Input the size of the array

printf("Enter the size of the array: ");

scanf("%d", &size);

int arr[size];

// Input array elements

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

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

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

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

}

// Call the function to perform Bubble Sort using pointers

bubbleSort(arr, size);

// Display the sorted array

printf("Sorted array:\n");

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

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

}

return 0;

}  
out put:

