//PROGRAM TO FIND THE SUM OF ELEMENTS IN AN ARRAY USING POINTERS//

#include <stdio.h>

int arraySum(int \*arr, int size) {

int sum = 0;

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

sum += \*(arr + i); // Accessing array elements using pointers

}

return sum;

}

int main() {

int size;

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

scanf("%d", &size);

int arr[size];

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

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

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

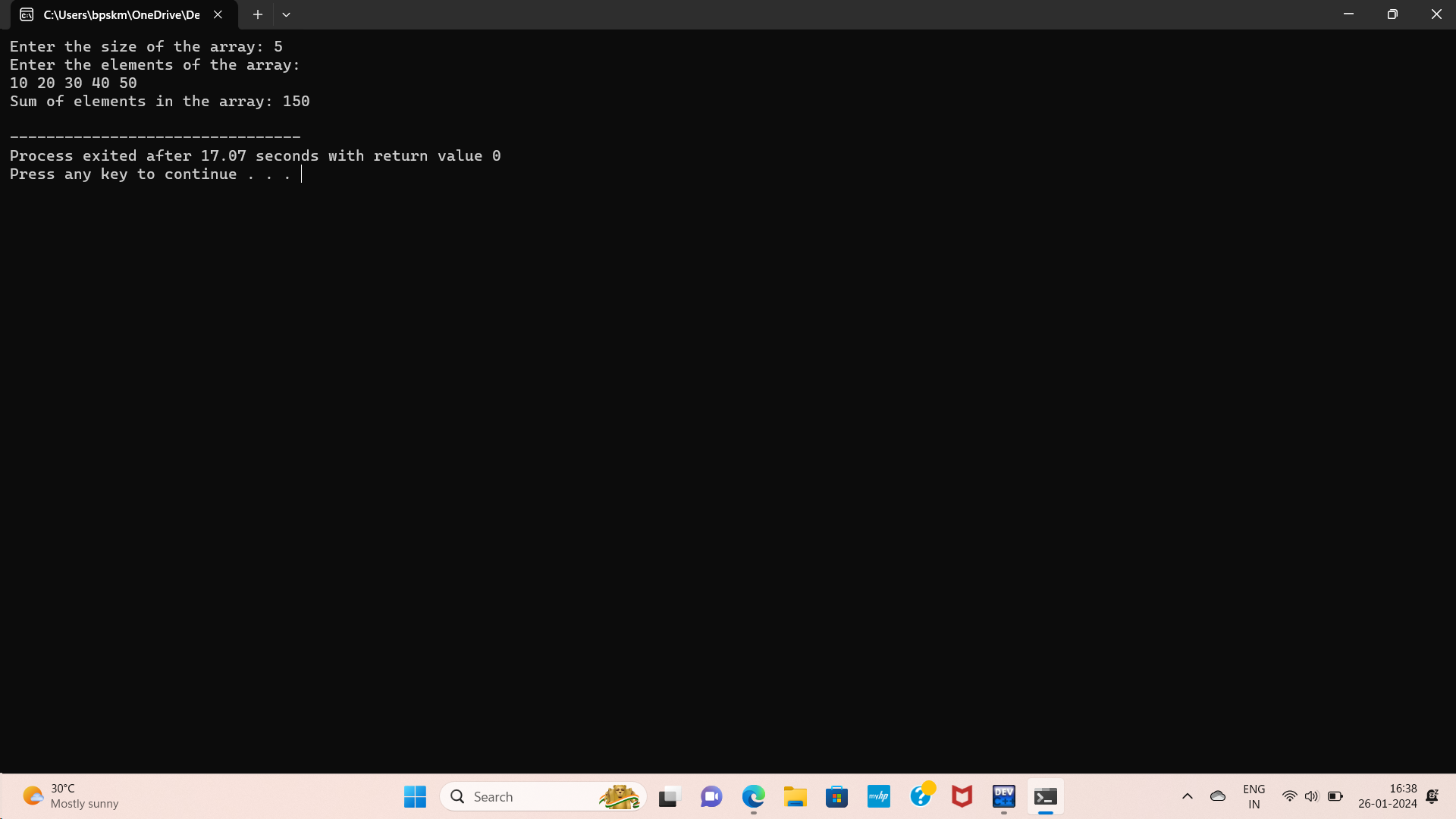
}

int sum = arraySum(arr, size);

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

return 0;

}



//PROGRAM TO SWAP TWO NUMBERS USING POINTERS//

#include <stdio.h>

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

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int num1, num2;

printf("Enter the first integer: ");

scanf("%d", &num1);

printf("Enter the second integer: ");

scanf("%d", &num2);

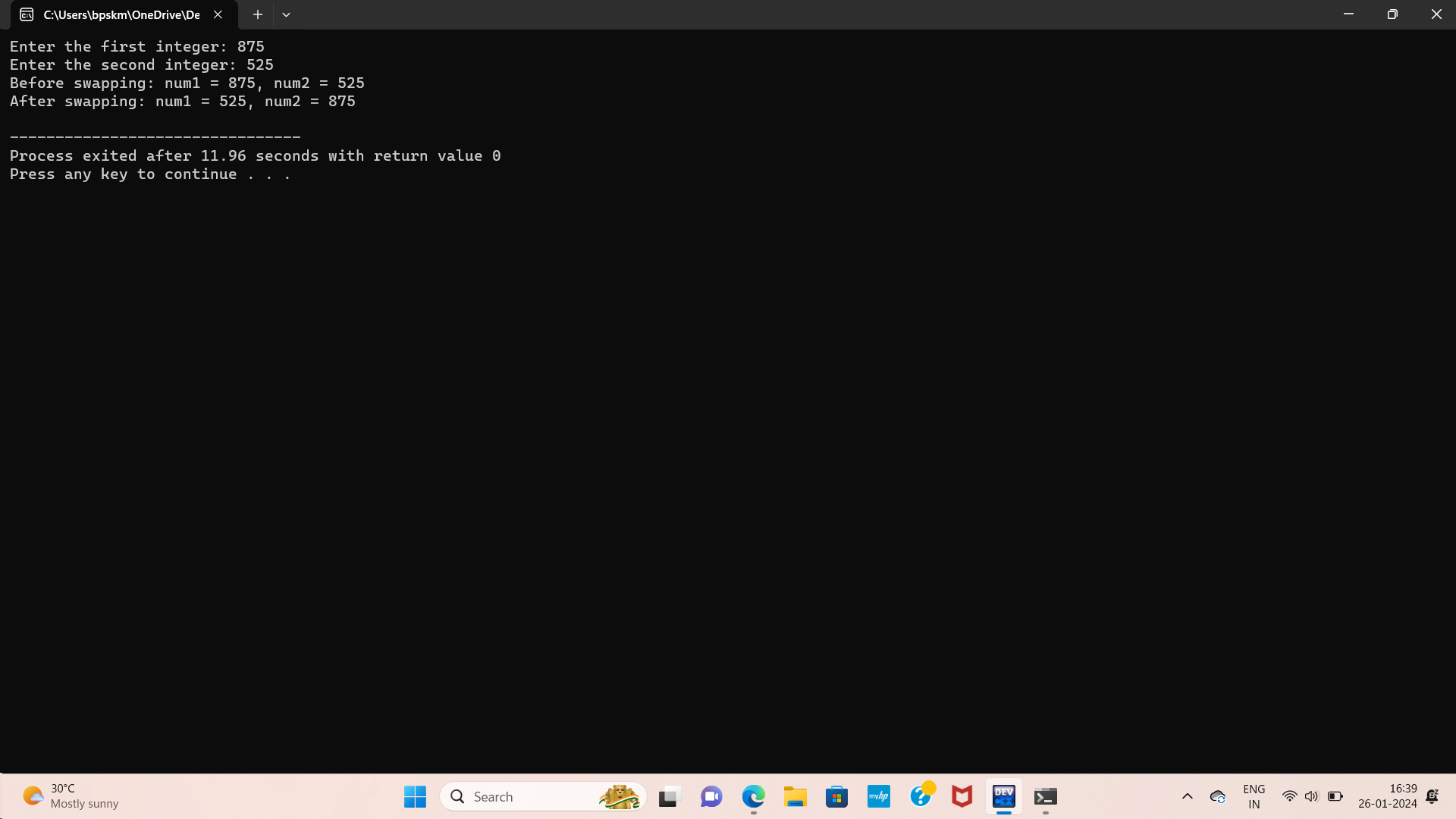
printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);

swap(&num1, &num2);

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

return 0;

}



//PROGRAM TO REVERSE A STRING USING POINTERS//

#include<stdio.h>

#include<string.h>

void reverseString(char \*str);

int main()

{

char S1[20];

printf("Enter the string:");

scanf("%s",&S1);

reverseString(S1);

printf("Reversed string:%s\n", S1);

return 0;

}

void reverseString(char \*str)

{

int length = strlen(str);

char \*a = str;

char \*b = str + length - 1;

while(a<b)

{

char temp = \*a;

\*a = \*b;

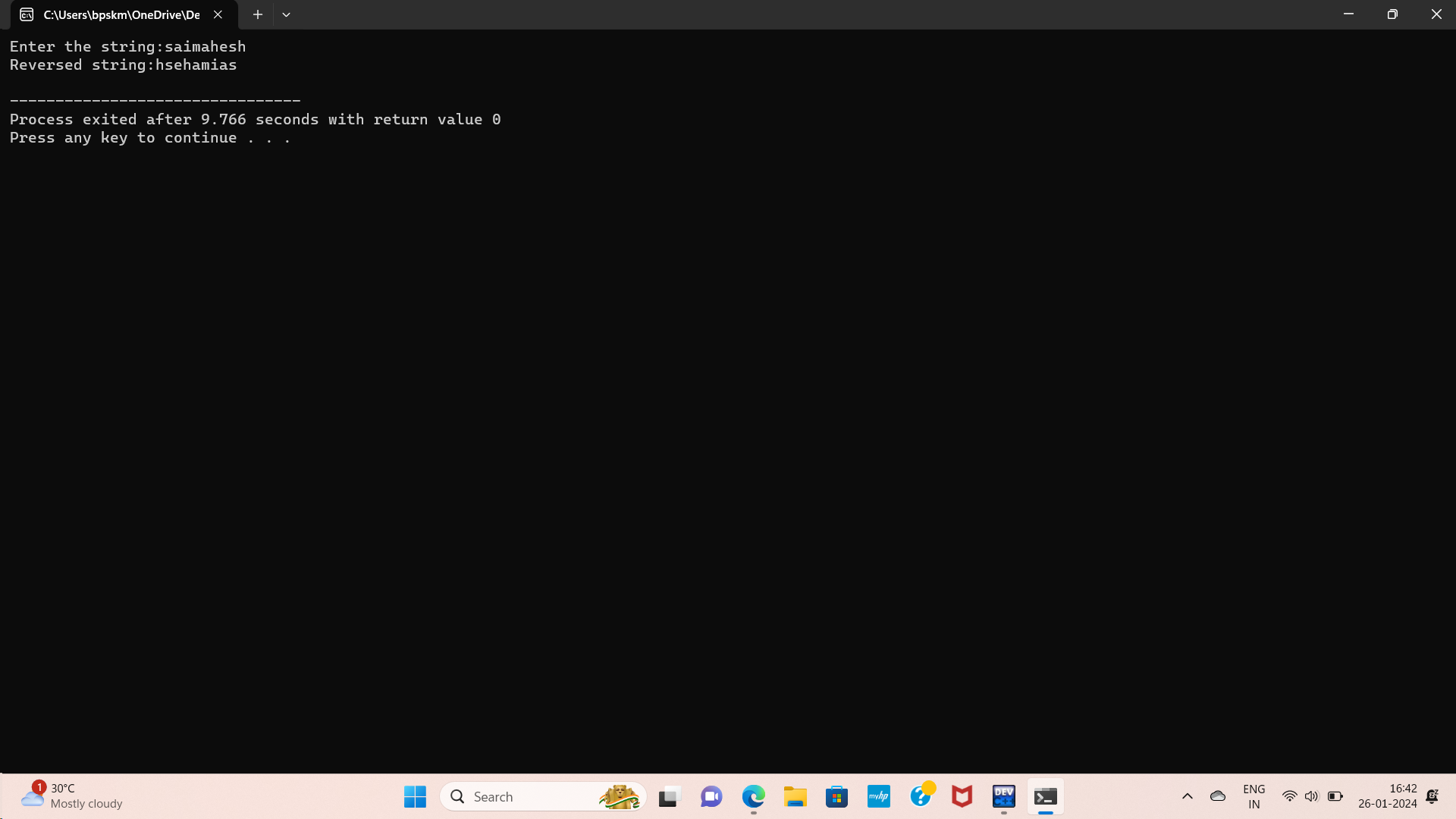
\*b = temp;

a++;

b--;

}

}



//PROGRAM TO CALCULATE THE POWER OF A NUMBER USING POINTERS TO FUNCTIONS//

#include <stdio.h>

// Function to calculate power

int power(int base, int exponent) {

int result = 1.0;

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

result \*= base;

}

return result;

}

// Function pointer type for power calculation

typedef int (\*PowerFunction)(int, int);

// Function that uses a function pointer to calculate power

double calculatePower(int base, int exponent, PowerFunction powFunc) {

return powFunc(base, exponent);

}

int main() {

int base, result;

int exponent;

printf("Enter the base:");

scanf("%d", &base);

printf("Enter the exponent:");

scanf("%d", &exponent);

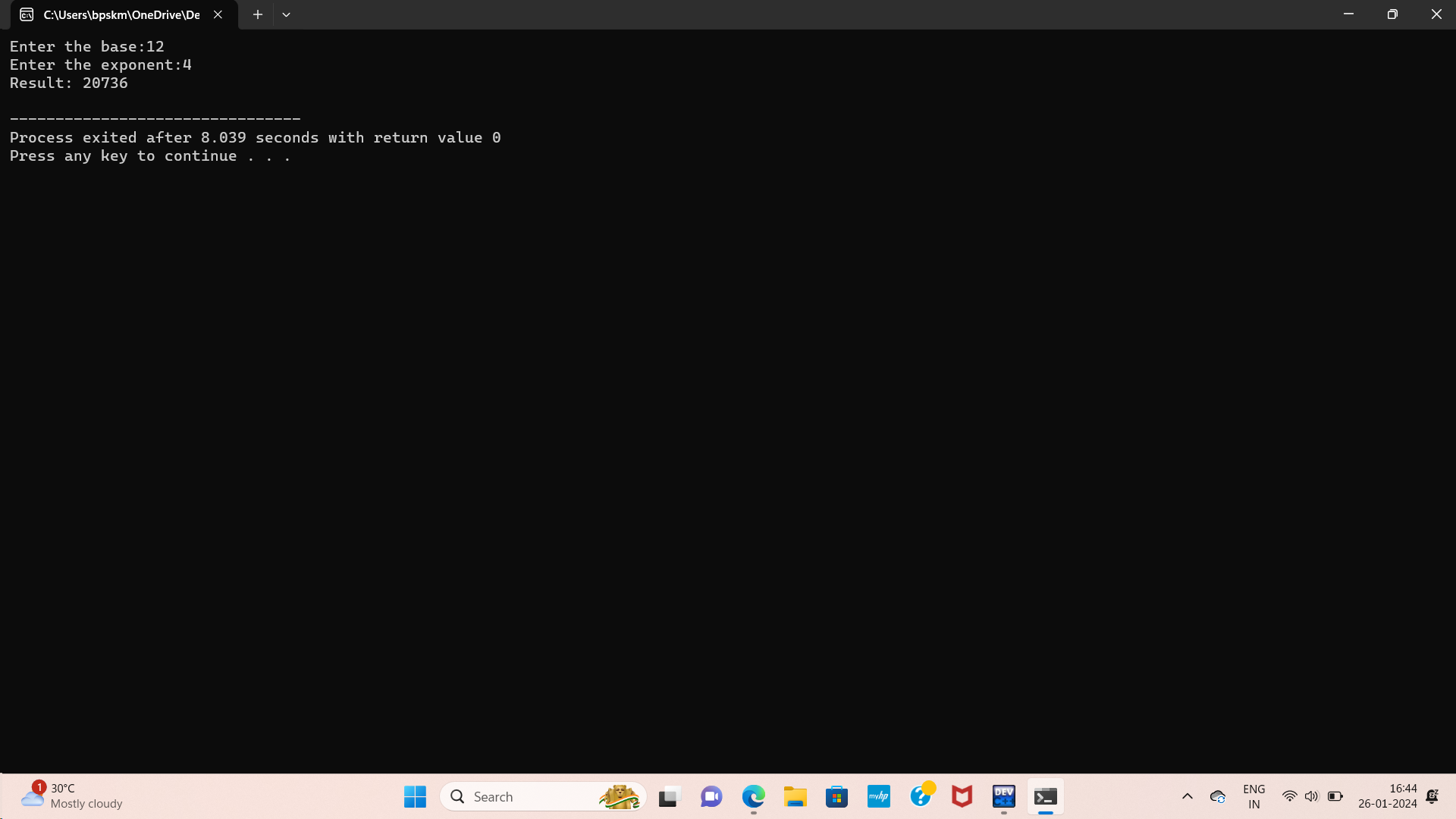
// Using the function pointer to call the power function

result = calculatePower(base, exponent, power);

printf("Result: %d\n", result);

return 0;

}



QUESTION::5

#include <stdio.h>

#include <stdlib.h>

int main()

{

int rows, columns;

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

scanf("%d", &rows);

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

scanf("%d", &columns);

// Dynamically allocate memory for the 2D array

int \*\*matrix = (int \*\*)malloc(rows \* sizeof(int \*));

if (matrix == NULL)

{

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

return 1;

}

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

{

matrix[i] = (int \*)malloc(columns \* sizeof(int));

if (matrix[i] == NULL)

{

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

for (int j = 0; j < i; j++)

{

free(matrix[j]);

}

free(matrix);

return 1;

}

}

// Input values into the 2D array

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

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

{

for (int j = 0; j < columns; j++)

{

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

}

}

// Display the 2D array

printf("Matrix:\n");

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

{

for (int j = 0; j < columns; j++)

{

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

}

printf("\n");

}

// Free allocated memory

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

{

free(matrix[i]);

}

free(matrix);

return 0;

}

