#include <stdio.h>

int main()

{

int arr[] = {1, 2, 3, 4, 5};

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

int n = 3;

printf("Original array: \n");

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

{

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

}

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

{

int j, last;

last = arr[length-1];

for(j = length-1; j > 0; j--)

{

arr[j] = arr[j-1];

}

arr[0] = last;

}

printf("\n");

printf("Array after right rotation: \n");

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

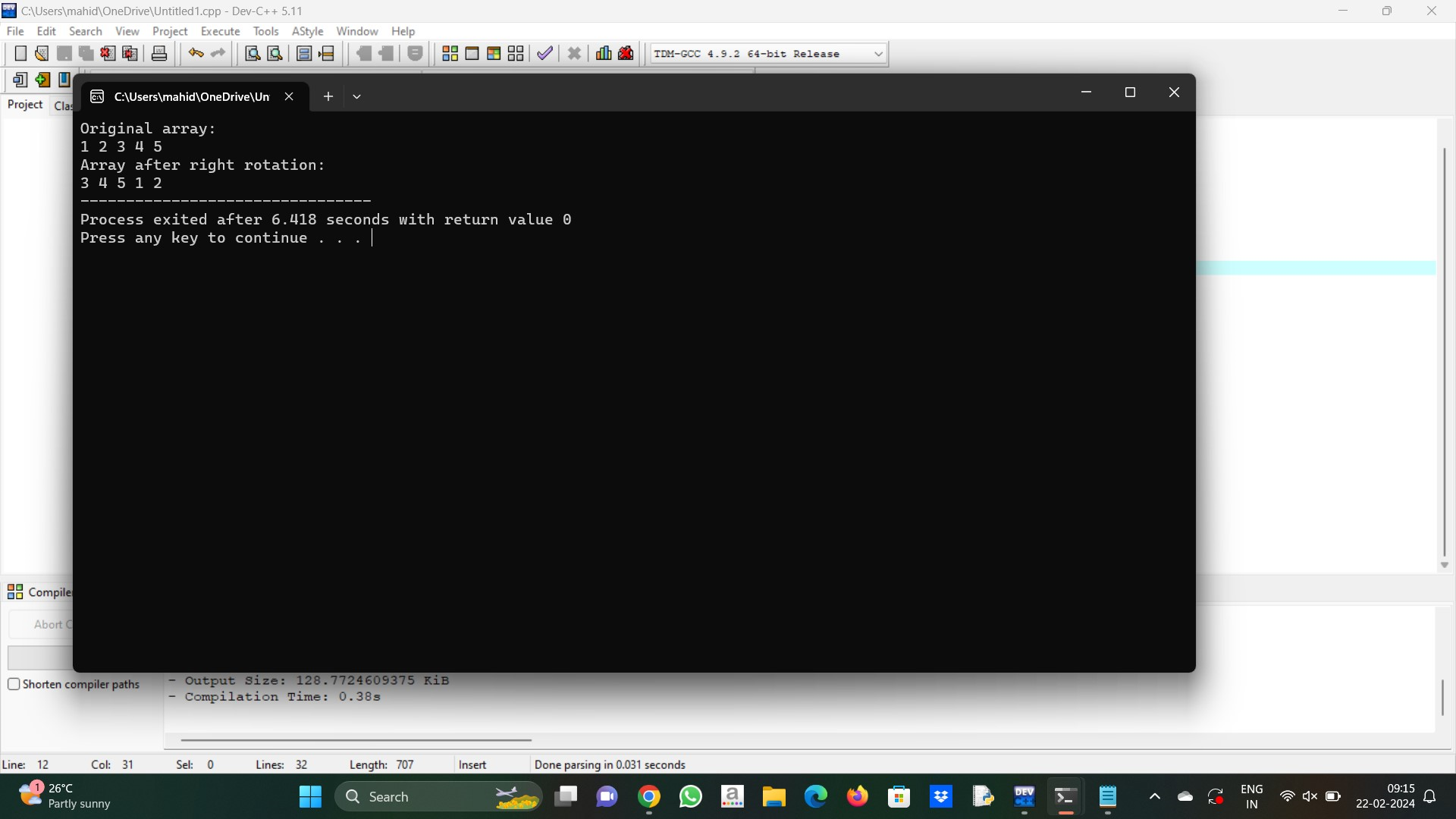
{

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

}

return 0;

}



2. #include<stdio.h>

int main()

{

int size,m=0,l=0;

printf("Type the length of the array\n");

scanf("%d",&size);

int array[size];

printf("type the elements of the array\n");

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

{

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

}

int largest=array[0];

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

{

int sum=0;

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

{

sum=sum+array[j];

if(sum>largest)

{

m=i;l=j;

largest=sum;

}

}

}

printf("\n The largest contigous subarray is");

for(int z=m;z<=l;z++)

{

printf(" %d ",array[z]);

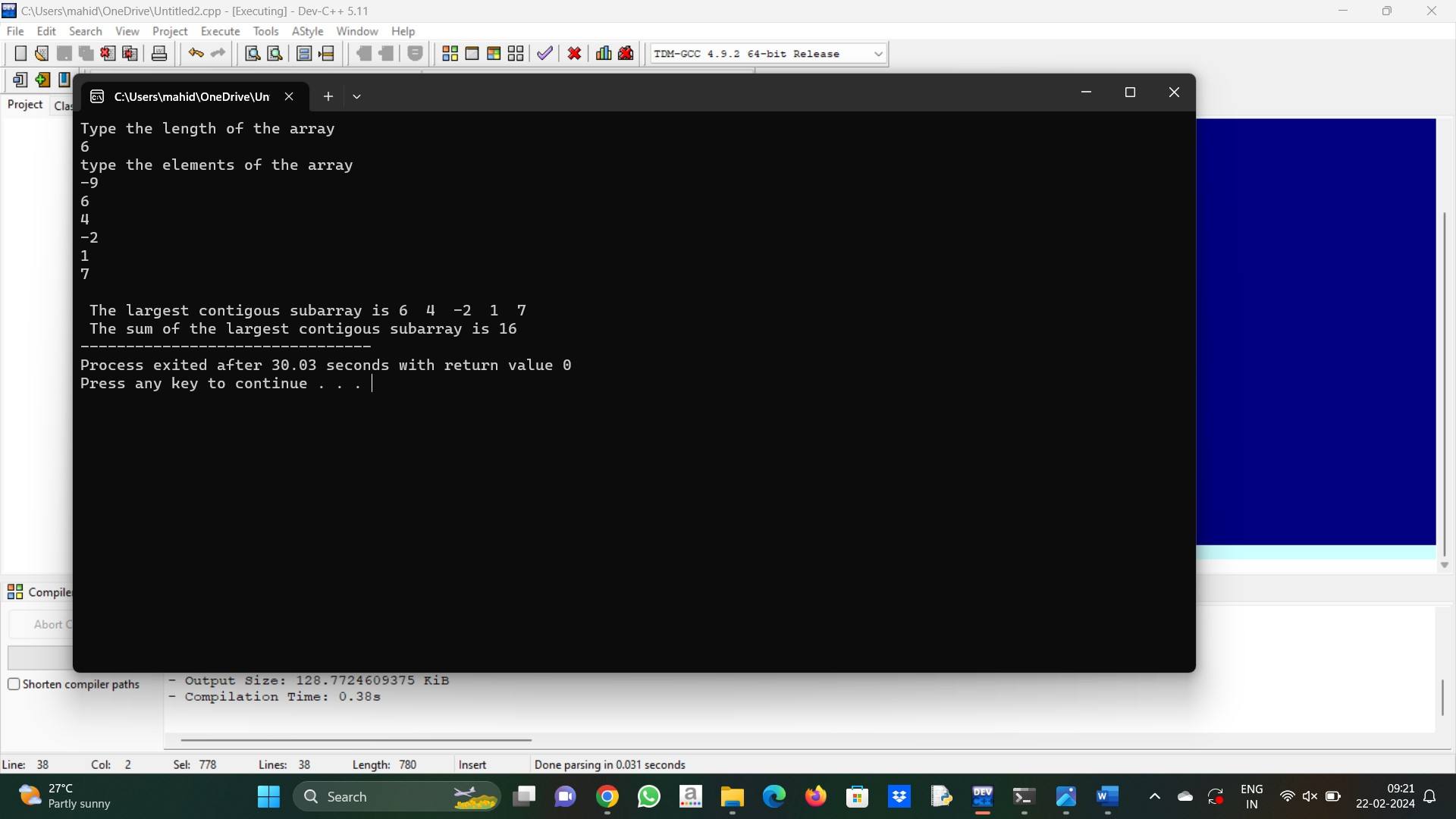
}

printf("\n The sum of the largest contigous subarray is");

printf(" %d",largest);

return 0;

}



3. #include<stdio.h>

#define N 10

int main()

{

int a[N], i, j = N, temp;

printf("Enter %d integer numbers\n", N);

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

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

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

{

if(a[i] % 2 != 0)

{

while(j > i)

{

j--;

if(a[j] % 2 == 0)

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

break;

}

}

}

}

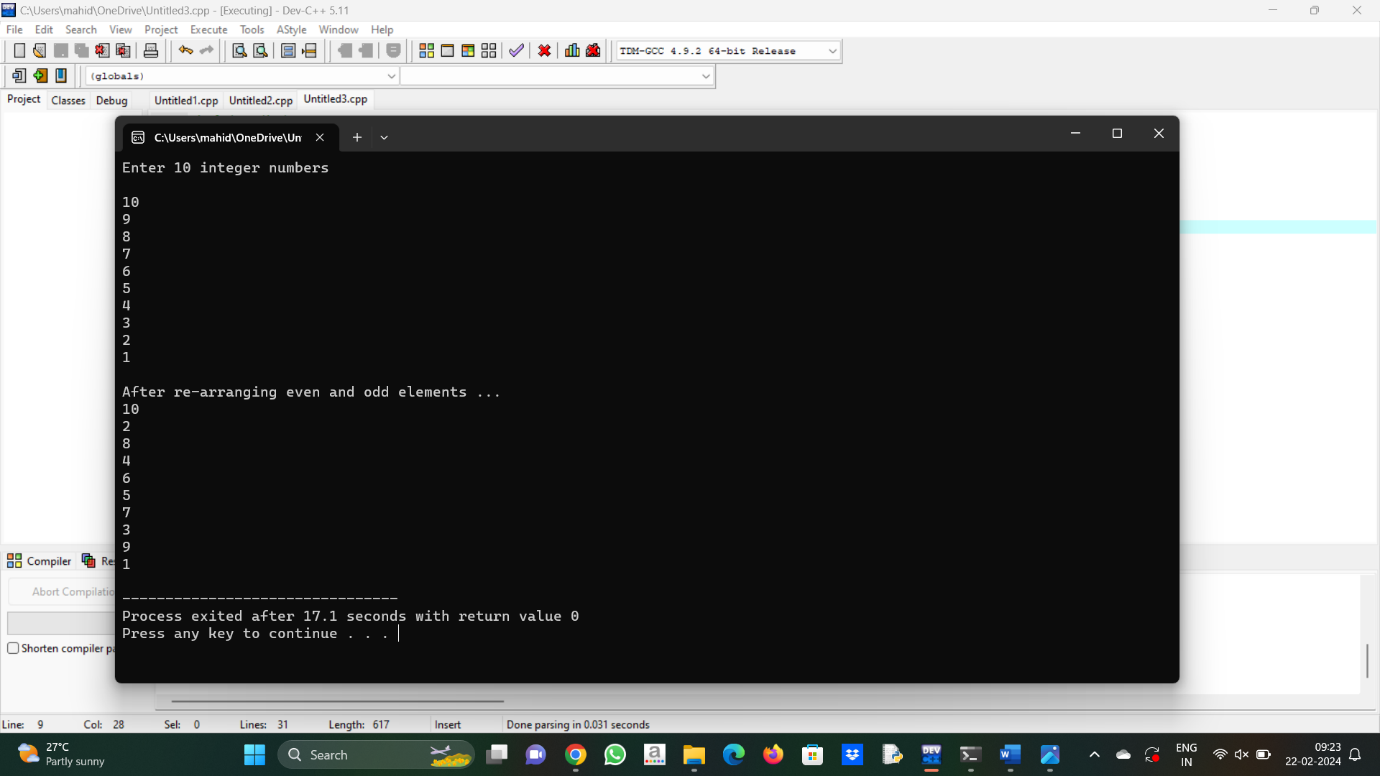
printf("\nAfter re-arranging even and odd elements ...\n");

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

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

return 0;

}



4. #include <stdio.h>

#include <limits.h>

void findLrgSubArray(int \*arr1, int arr\_size)

{

int i, j, sum = 0, lrgSize = INT\_MIN, left;

for (i = 0; i < arr\_size - 1; i++)

{

sum = arr1[i] ? 1 : -1;

for (j = i + 1; j < arr\_size; j++)

{

if (arr1[j] == 1)

sum += 1;

else

sum += -1;

if (sum == 0 && (lrgSize < j - i + 1))

{

lrgSize = j - i + 1;

left = i;

}

}

}

if (lrgSize == INT\_MIN)

{

printf("No such subarray found from the given array.");

}

else

{

printf("Subarray found from the index %d to %d", left, left + lrgSize - 1);

}

}

int main()

{

int i, arr1[] = {0, 1, 0, 0, 1, 1, 0, 1, 1, 1};

int n = sizeof(arr1) / sizeof(arr1[0]);

printf("The given array is : ");

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

{

printf("%d ", arr1[i]);

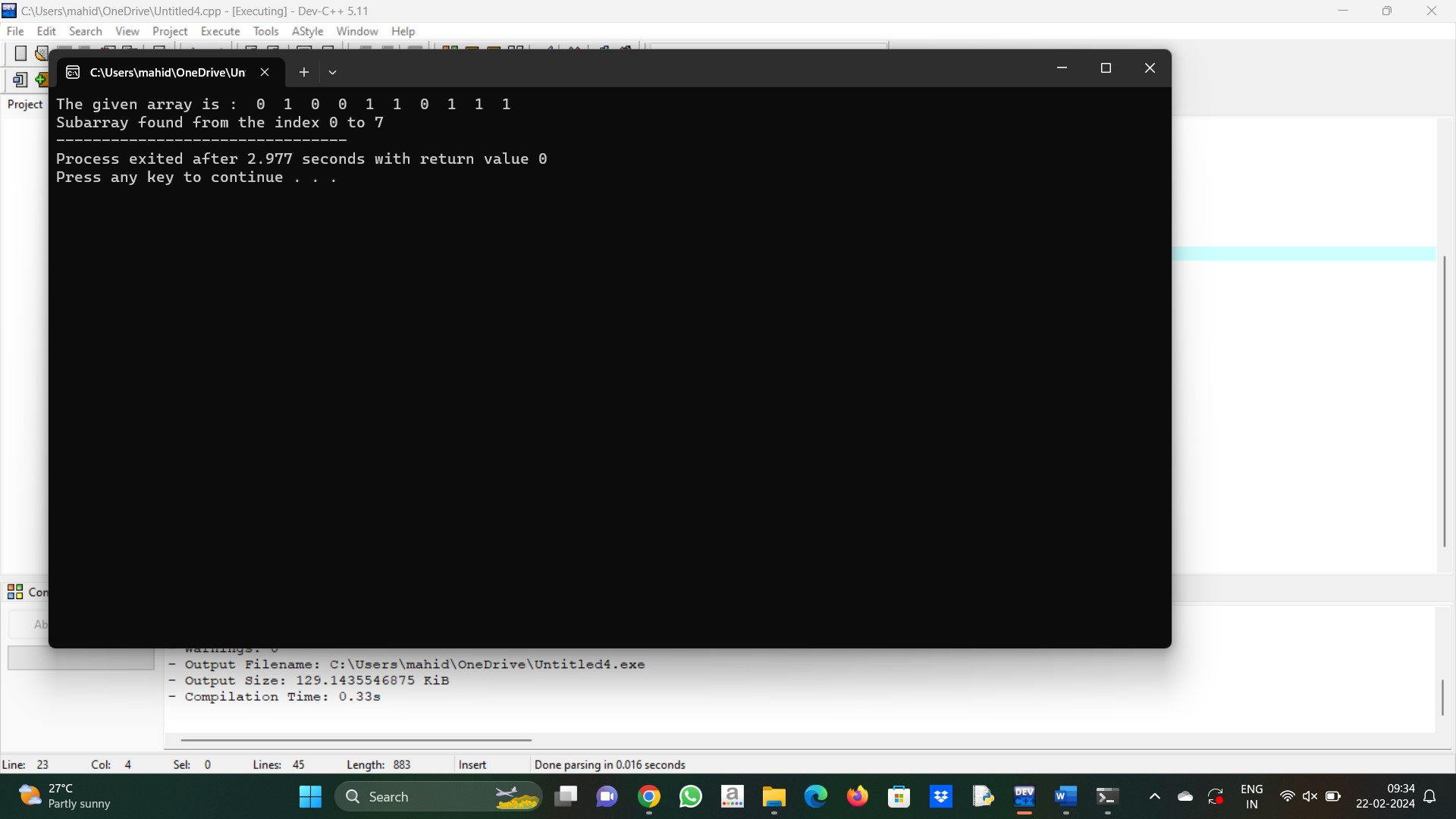
}

printf("\n");

findLrgSubArray(arr1, n);

return 0;

}



5. #include <stdbool.h>

#include <stdio.h>

int firstMissingPos(int A[], int n)

{

bool present[n + 1];

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

present[i] = false;

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

if (A[i] > 0 && A[i] <= n)

present[A[i]] = true;

}

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

if (!present[i])

return i;

return n + 1;

}

int main()

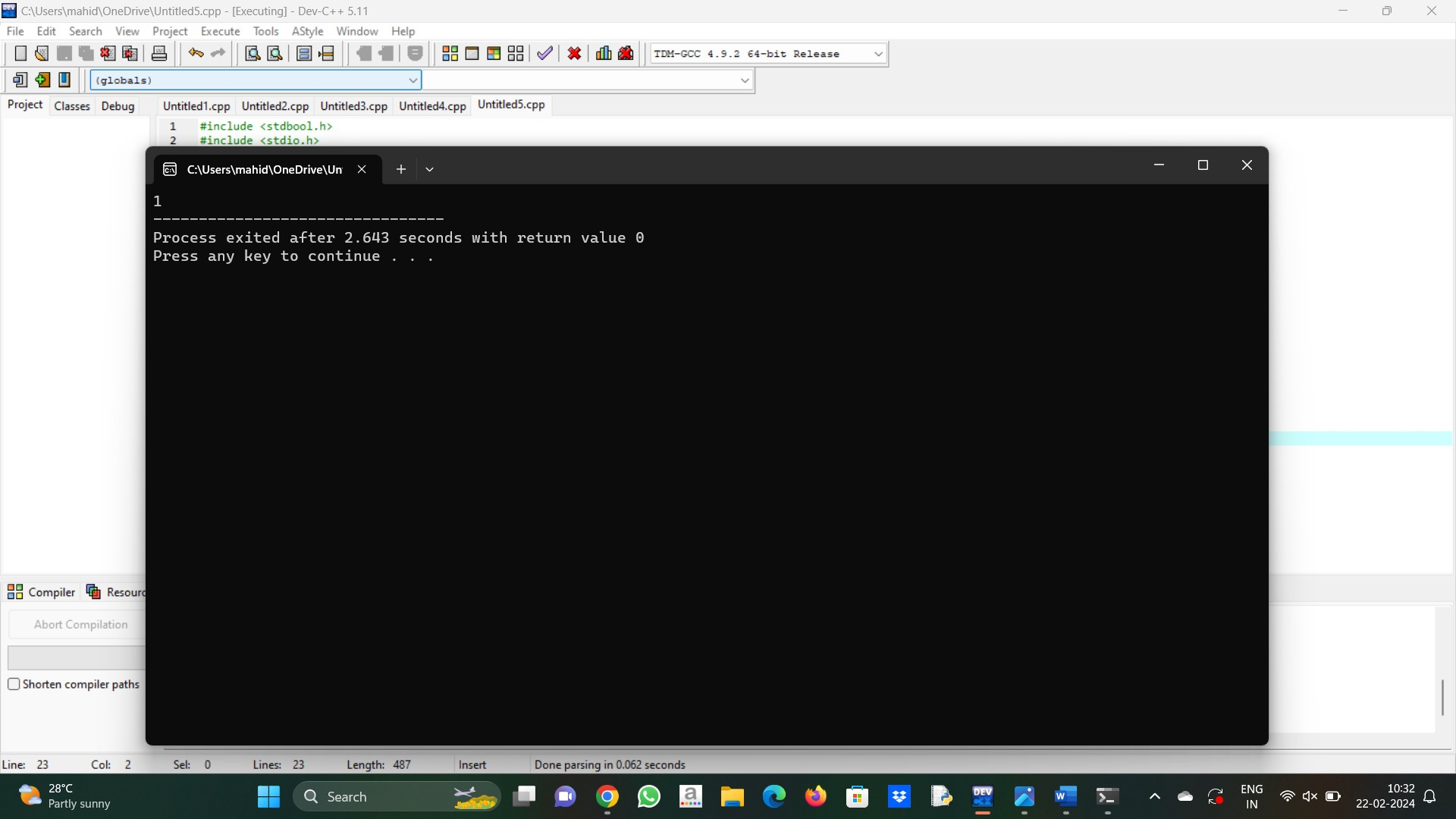
{

int arr[] = { 0, 10, 2, -10, -20 };

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

printf("%d", firstMissingPos(arr, size));

}



6. #include <stdio.h>

#include <limits.h>

void findMaximumProduct(int arr[], int n)

{

if (n < 2)

{

return;

}

int max\_product = INT\_MIN;

int max\_i, max\_j;

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

{

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

{

if (max\_product < arr[i] \* arr[j])

{

max\_product = arr[i] \* arr[j];

max\_i = i, max\_j = j;

}

}

}

printf("Pair is (%d, %d)", arr[max\_i], arr[max\_j]);

}

int main()

{

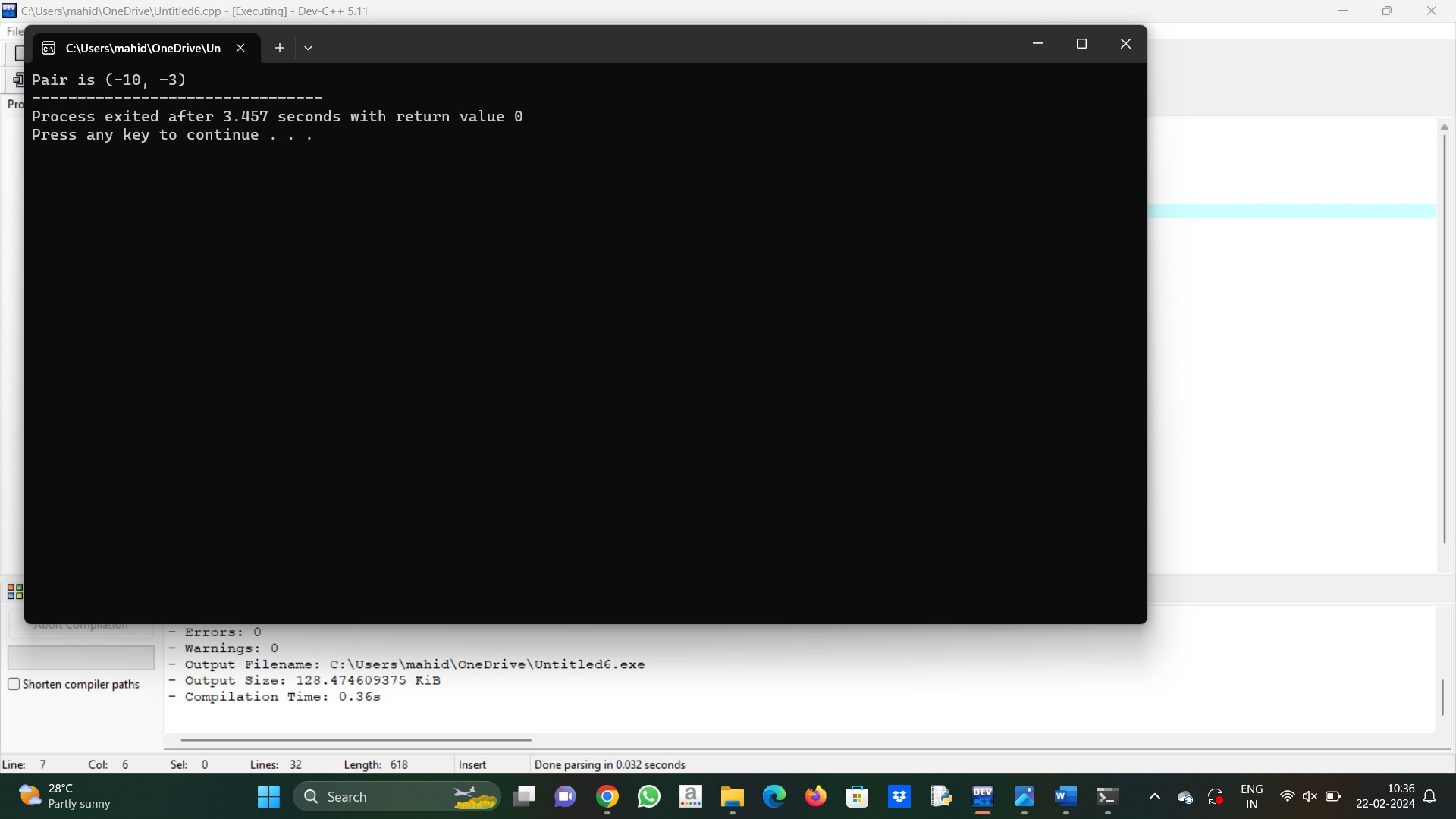
int arr[] = { -10, -3, 5, 6, -2 };

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

findMaximumProduct(arr, n);

return 0;

}



7. #include <stdio.h>

void print(int arr1[], int i, int j)

{

printf("[%d..%d] -- { ", i, j);

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

{

printf("%d ", arr1[k]);

}

printf("}\n");

}

void PickSubarrayFromArray(int arr1[], int n, int sum)

{

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

{

int sum\_upto = 0;

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

{

sum\_upto += arr1[j];

if (sum\_upto == sum)

{

print(arr1, i, j);

}

}

}

}

int main()

{

int arr1[] = { 3, 4, -7, 1, 3, 3, 1, -4 };

int sum = 7;

int ctr = sizeof(arr1) / sizeof(arr1[0]);

int i;

printf("The given array is : ");

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

{

printf("%d ", arr1[i]);

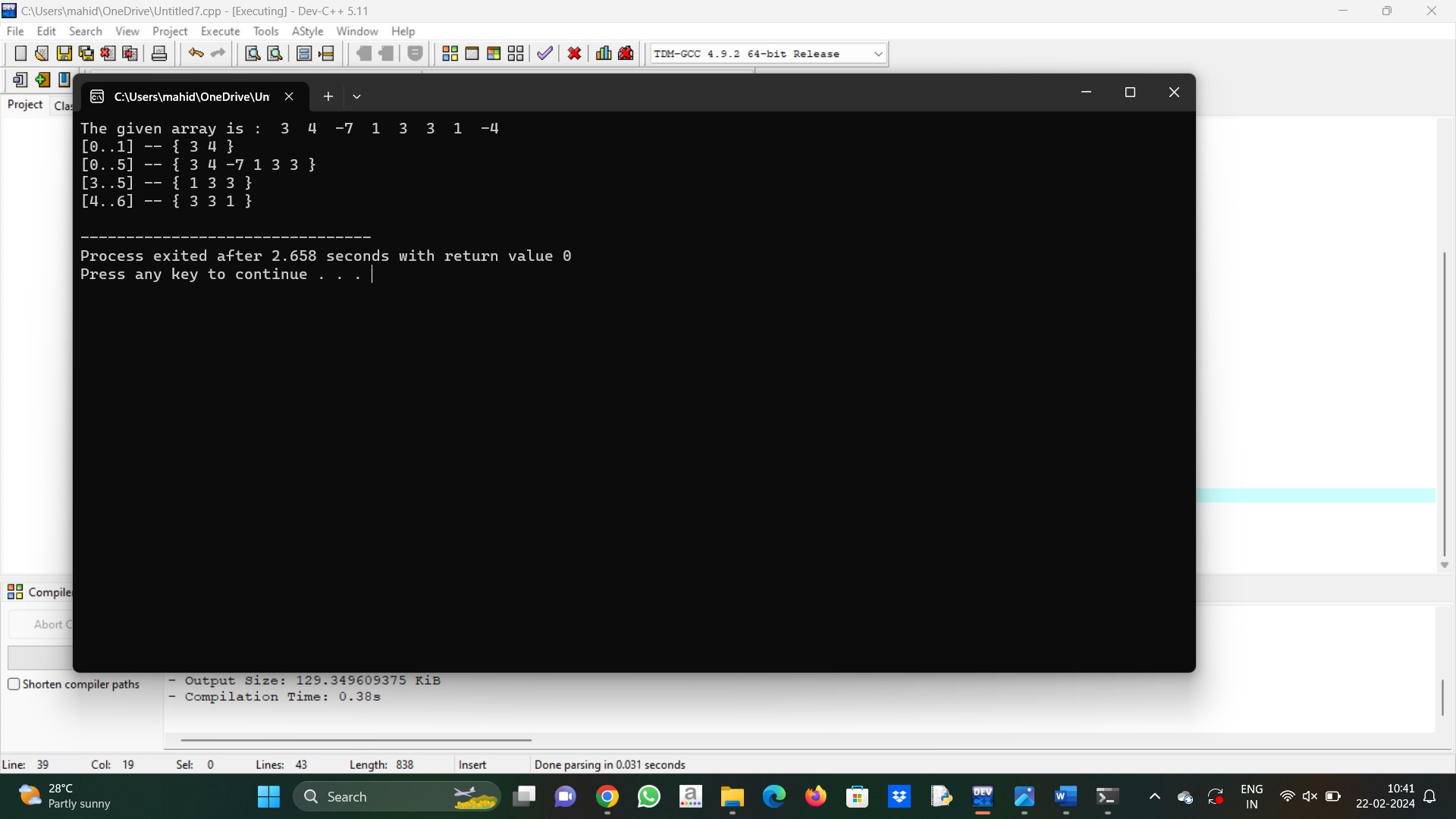
}

printf("\n");

PickSubarrayFromArray(arr1, ctr, sum);

return 0;

}



8. #include <stdio.h>

#include <stdbool.h>

#define MAX 100

bool findSaddlePoint(int mat[MAX][MAX], int n)

{

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

{

int min\_row = mat[i][0], col\_ind = 0;

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

{

if (min\_row > mat[i][j])

{

min\_row = mat[i][j];

col\_ind = j;

}

}

int k;

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

if (min\_row < mat[k][col\_ind])

break;

if (k == n)

{

printf("Value of Saddle Point %d",min\_row);

return true;

}

}

return false;

}

int main()

{

int mat[MAX][MAX] = {{1, 2, 3},

{4, 5, 6},

{7, 8, 9}};

int n = 3;

if (findSaddlePoint(mat, n) == false)

printf("No Saddle Point ");

return 0;

}

