8 to 9 programs:

1.Given an array of integers, find the maximum difference between any two elements in the array.

Program:

#include<stdio.h>

int maxDiff(int arr[], int arr\_size)

{

int max\_diff = arr[1] - arr[0];

int i, j;

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

{

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

{

if (arr[j] - arr[i] > max\_diff)

max\_diff = arr[j] - arr[i];

}

}

return max\_diff;

}

int main()

{

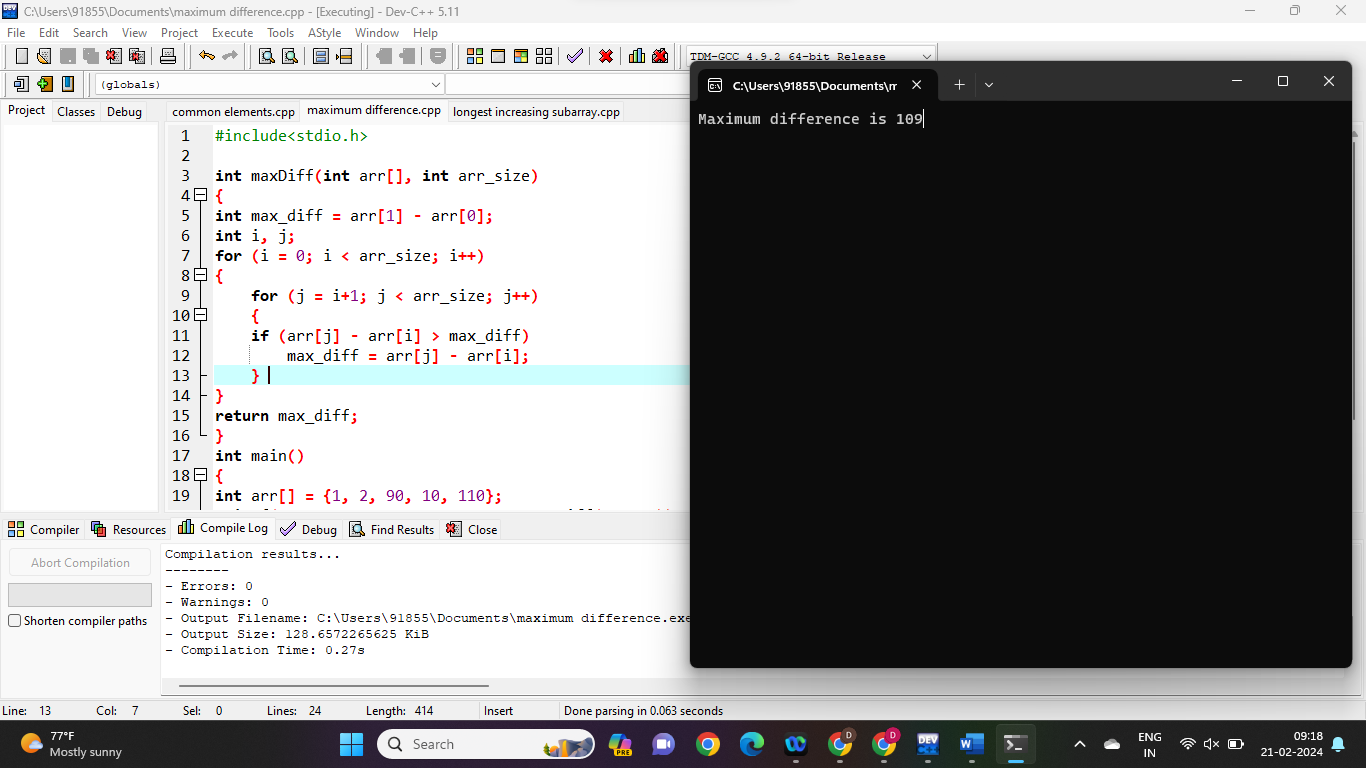
int arr[] = {1, 2, 90, 10, 110};

printf("Maximum difference is %d", maxDiff(arr, 5));

getchar();

return 0;

}



2. Given an array of integers, find the longest increasing subarray.

Program:

#include <stdio.h>

#include <stdlib.h>

int \_lis(int arr[], int n, int\* max\_ref)

{

if (n == 1)

return 1;

int res, max\_ending\_here = 1;

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

res = \_lis(arr, i, max\_ref);

if (arr[i - 1] < arr[n - 1]

&& res + 1 > max\_ending\_here)

max\_ending\_here = res + 1;

}

if (\*max\_ref < max\_ending\_here)

\*max\_ref = max\_ending\_here;

return max\_ending\_here;

}

int lis(int arr[], int n)

{

int max = 1;

\_lis(arr, n, &max);

return max;

}

int main()

{

int arr[] = { 10, 22, 9, 33, 21, 50, 41, 60 };

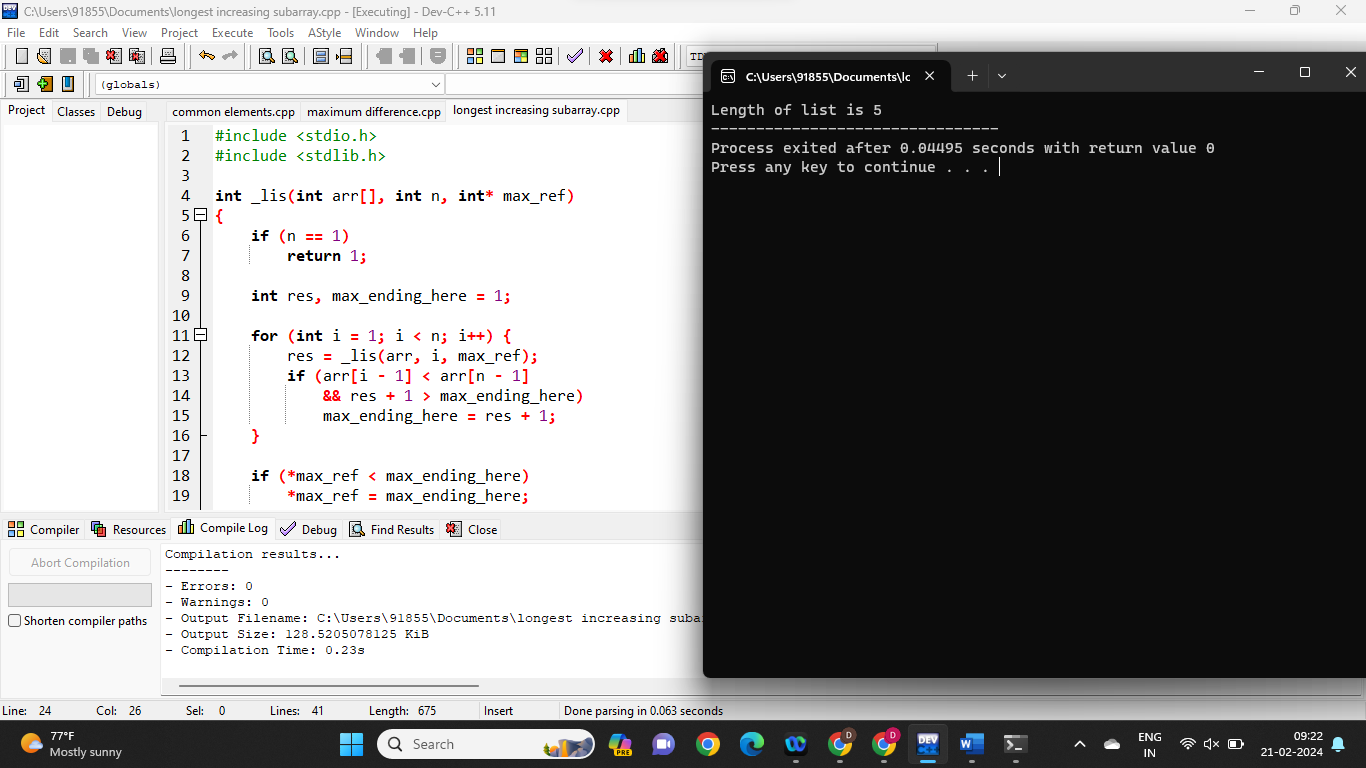
int n = sizeof(arr) / sizeof(arr[0]);

printf("Length of list is %d", lis(arr, n));

return 0;

}

Output:



3. Given two arrays of integers, find the common elements between them

Program:

#include <stdio.h>

int main()

{

int array1[] = { 8, 2, 3, 4, 5, 6, 7, 1 };

int array2[] = { 4, 5, 7, 11, 6, 1 };

int i, j, flag, x, k = 0;

int result[100];

printf("Common elements are: ");

for (i = 0; i < sizeof(array1) / 4; i++) {

for (j = 0; j < sizeof(array2) / 4; j++) {

if (array1[i] == array2[j]) {

flag = 0;

for (x = 0; x < k; x++) {

if (result[x] == array1[i]) {

flag++;

}

}

if (flag == 0) {

result[k] = array1[i];

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

k++;

}

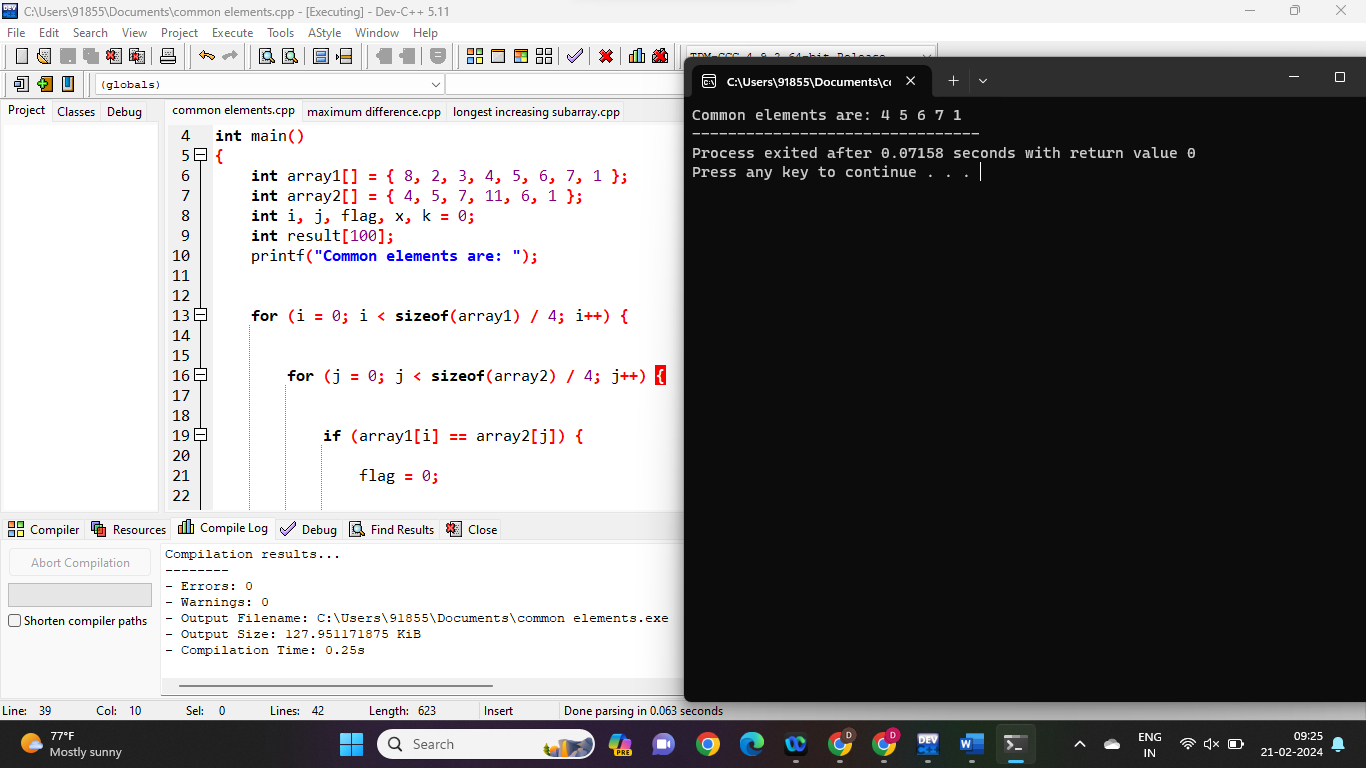
}

}

}

}

Output:



9 to 10 programs:

4.Given an array of integers, find the element that appears more than n/2 times (where n is the size of the array).

Program:

#include <stdio.h>

int main()

{

int arr[] = { 20, 25, 25, 25, 25, 42, 67 };

int item = 25;

int loop = 0;

int count = 0;

int size = 0;

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

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

if (item == arr[loop])

count = count + 1;

}

if (count > (size / 2))

printf("The number %d appears more than %d times in arr[]\n", item, size / 2);

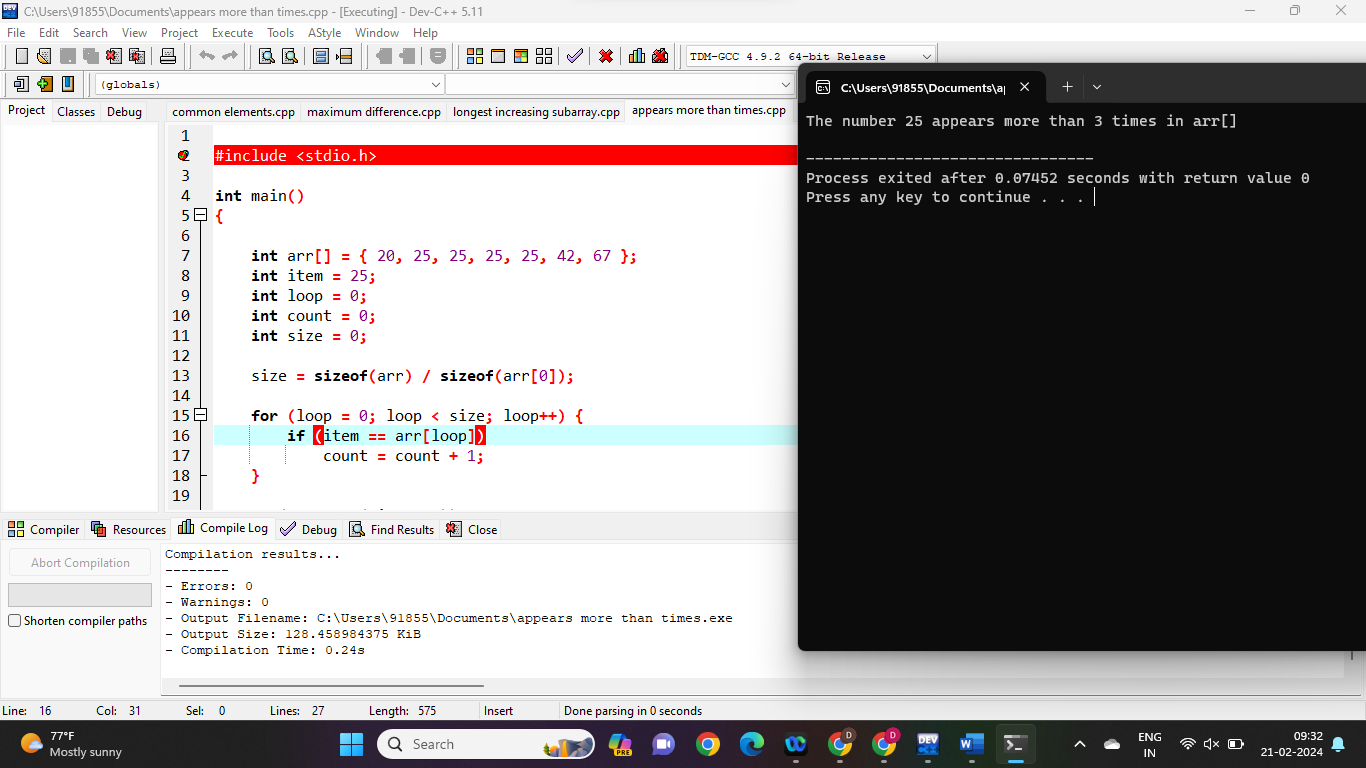
else

printf("The number %d does not appear more than %d times in arr[]\n", item, size / 2);

return 0;

}

Output:



5. Given an array of integers, rearrange the elements in such a way that all the negative elements come before the positive elements

Program:

#include<stdio.h>

#define ARRAY\_SIZE(a) sizeof(a)/sizeof(a[0])

void swap(int \*s1,int \*s2)

{

int temp = \*s1;

\*s1 = \*s2;

\*s2 = temp;

}

void segregateElements(int arr[],int n)

{

int i=0,j=0;

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

{

if(arr[i] < 0)

{

if(i != j)

{

swap(&arr[i],&arr[j]);

}

j++;

}

}

}

int main()

{

int arr[] = {-1,-2,-3,4,5,6,-7,8,9};

int arr\_size = ARRAY\_SIZE(arr);

int i = 0;

segregateElements(arr,arr\_size);

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

{

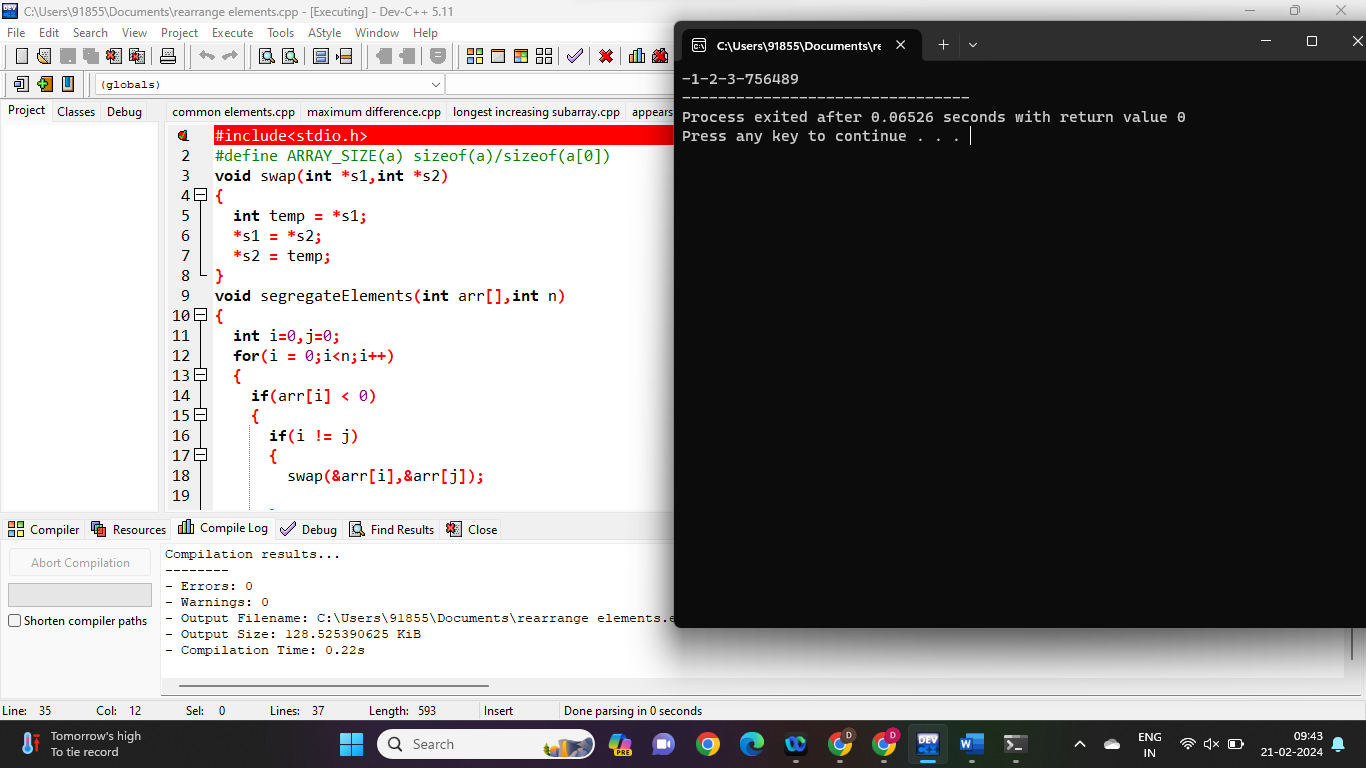
printf("%d",arr[i]);

}

return 0;

}

Output:



6. Given an array of integers, find the majority element (an element that appears more than n/2 times, where n is the size of the array) if it exists

Program:

#include<stdio.h>

#include<stdbool.h>

bool ismajority(int arr[],int n, int x)

{

int i;

int last\_index = n%2? (n/2+1): (n/2);

for(i=0;i<last\_index;i++)

{

if (arr[i] == x && arr[i+n/2]==x)

return 1;

}

return 0;

}

int main()

{

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

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

int x =4;

if(ismajority(arr,n,x))

printf("%d appears more than %d times in arr[]",x,n/2);

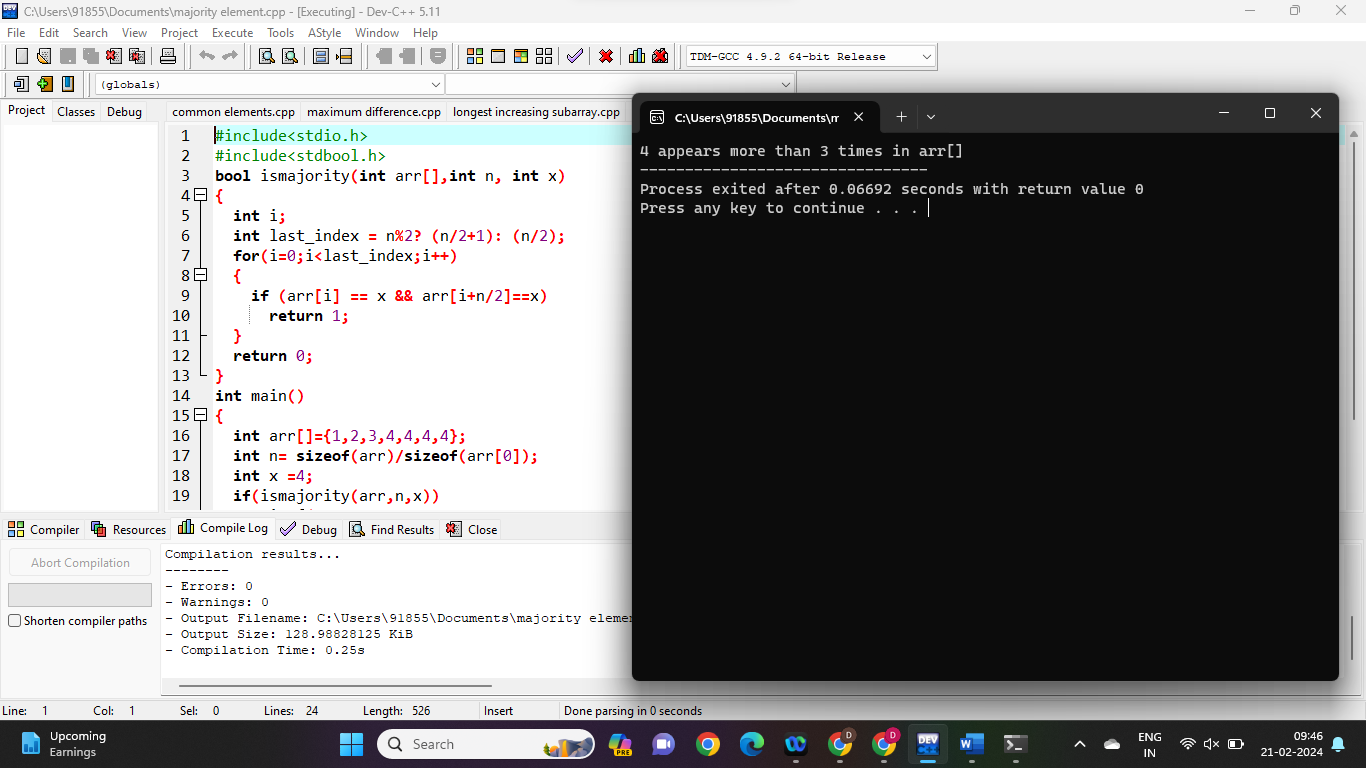
else

printf("%d does not appears more than %d times in arr[]",x,n/2);

return 0;

}

Output:



10 to 11 programs:

7. Given an array of integers, find the subarray with the largest sum.

Program:

#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]);

}

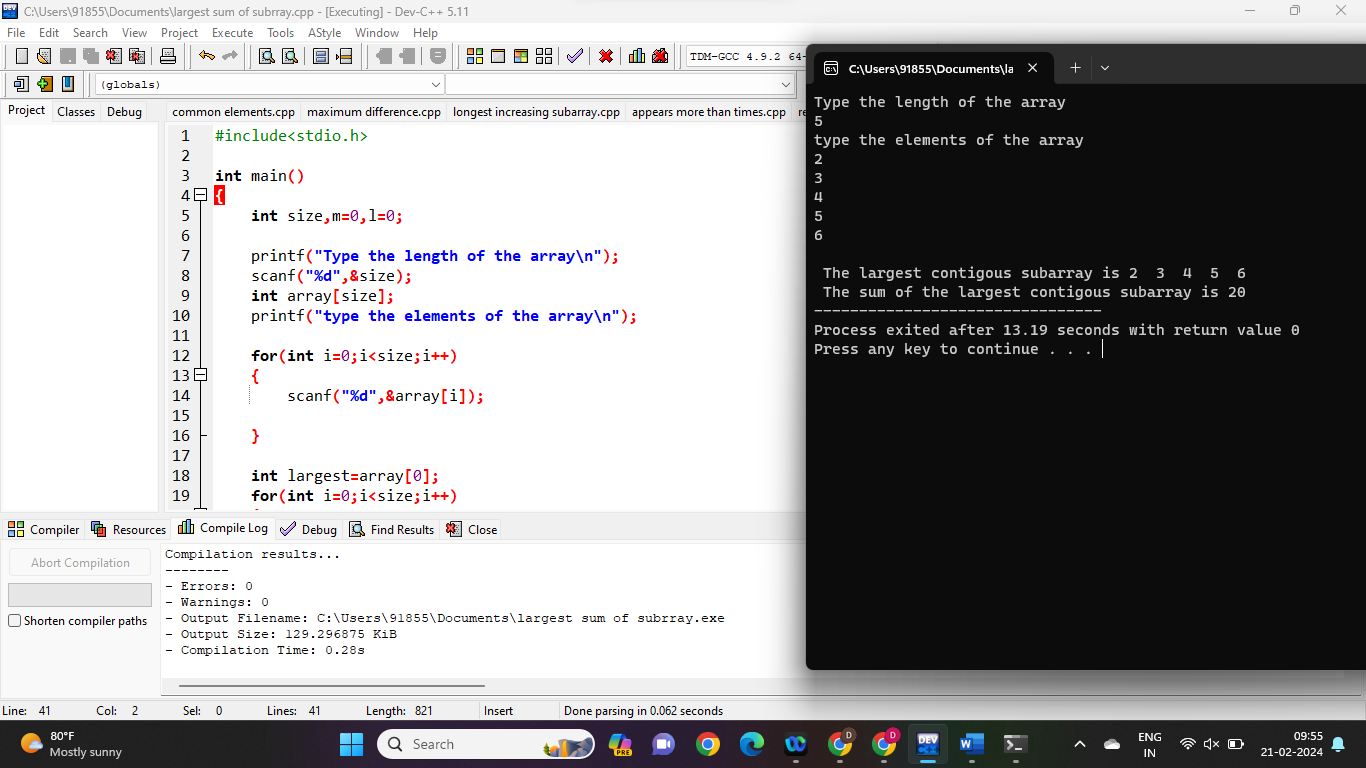
printf("\n The sum of the largest contigous subarray is");

printf(" %d",largest);

return 0;

}

Output:



8. Given an array of integers, rotate the array by k positions to the right.

Program:

#include <stdio.h>

int main()

{

int arr[] = { 1, 3, 5, 7, 9, 11 };

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

int k = 3;

k = k % n;

int i, j;

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

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

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

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

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

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

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

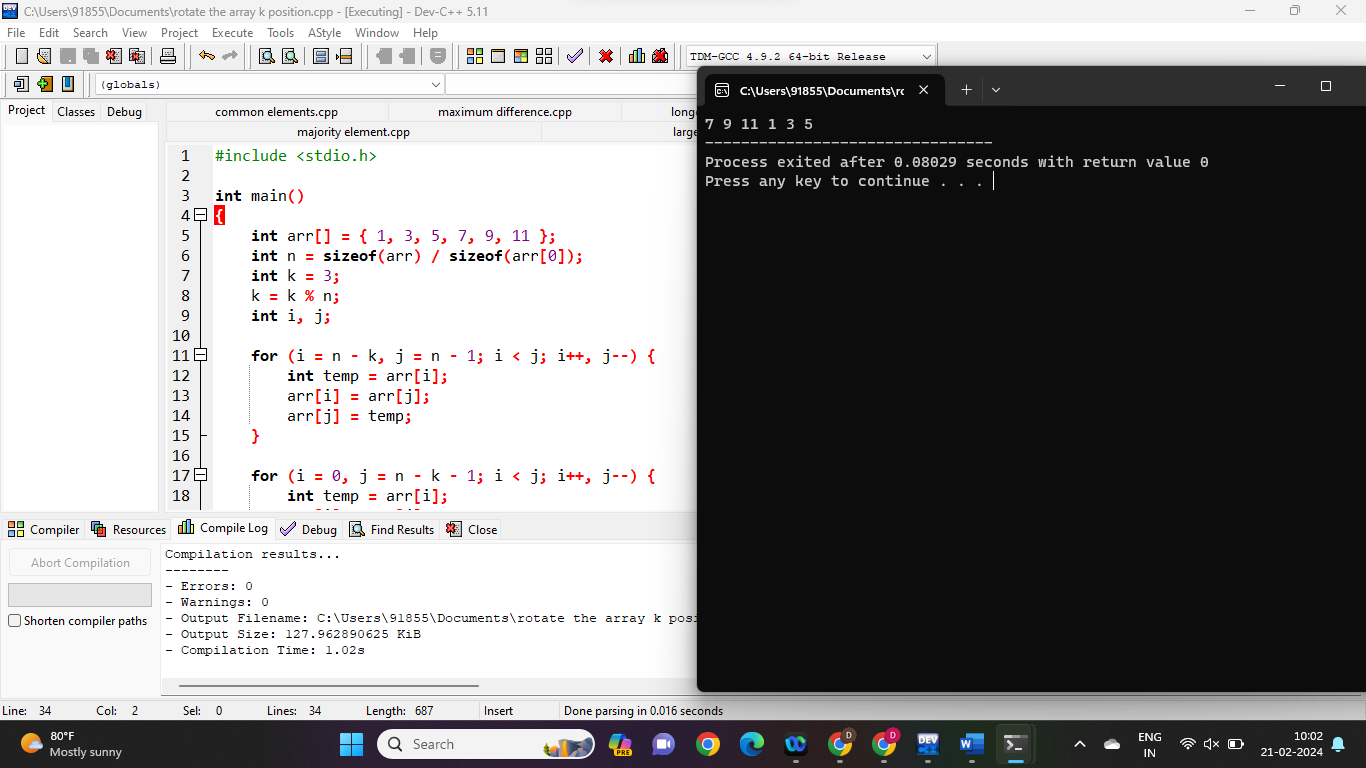
printf("%d ", arr[i]);

}

return 0;

}

Output:



9. Given an array of integers, rearrange the array in such a way that all the even elements come before the odd elements

Program:

#include <stdio.h>

#include <stdlib.h>

int cmpfunc(const void\* a, const void\* b)

{

return (\*(int\*)a - \*(int\*)b);

}

void assign(int a[], int n)

{

qsort(a, n, sizeof(int), cmpfunc);

int ans[n];

int p = 0, q = n - 1;

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

if ((i + 1) % 2 == 0)

ans[i] = a[q--];

else

ans[i] = a[p++];

}

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

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

}

int main()

{

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

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

assign(A, n);

return 0;

}

Output:

