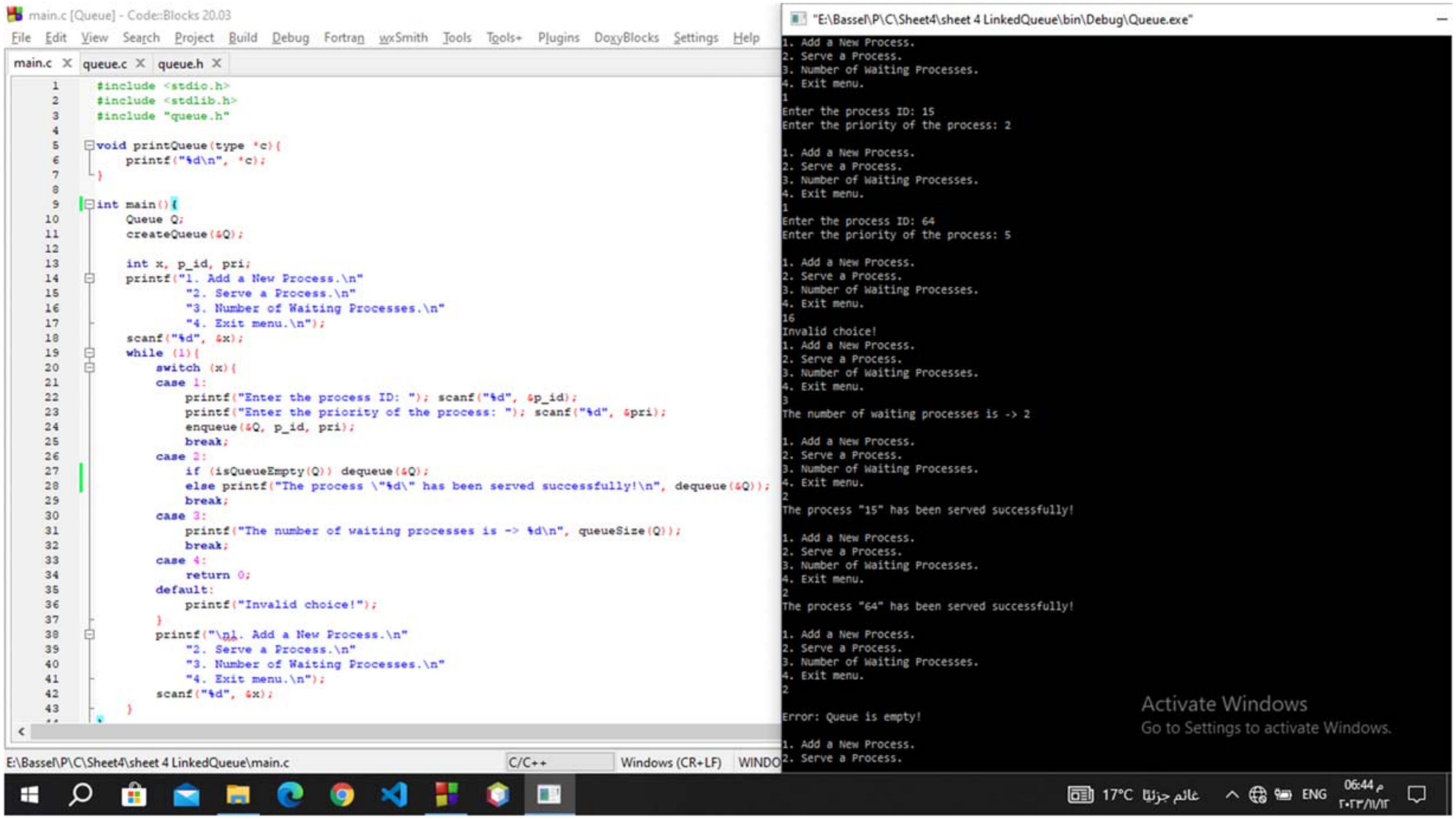
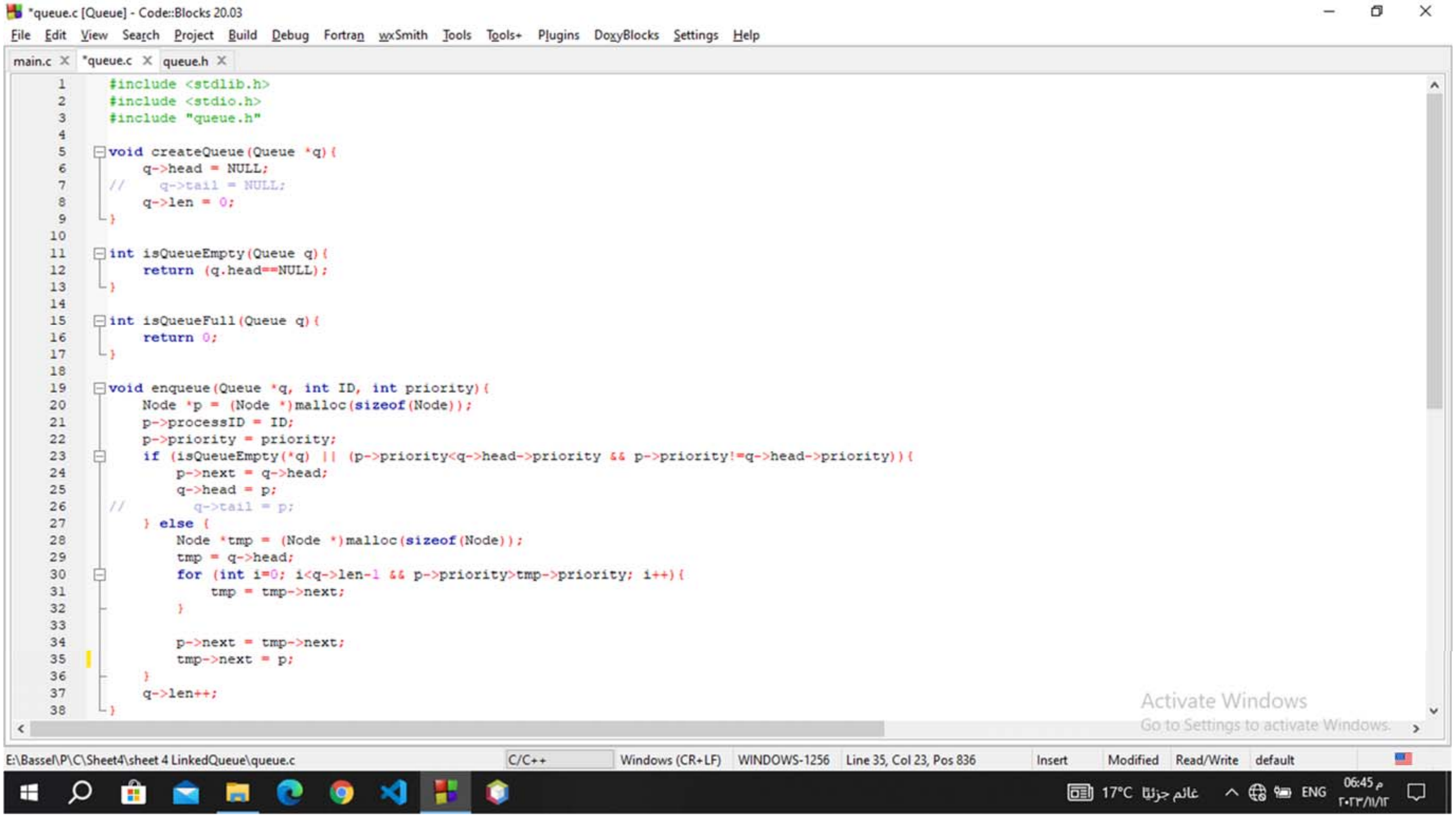
Name : Ahmed Hassan Ahmed Emara ID : 20220017

Q1:

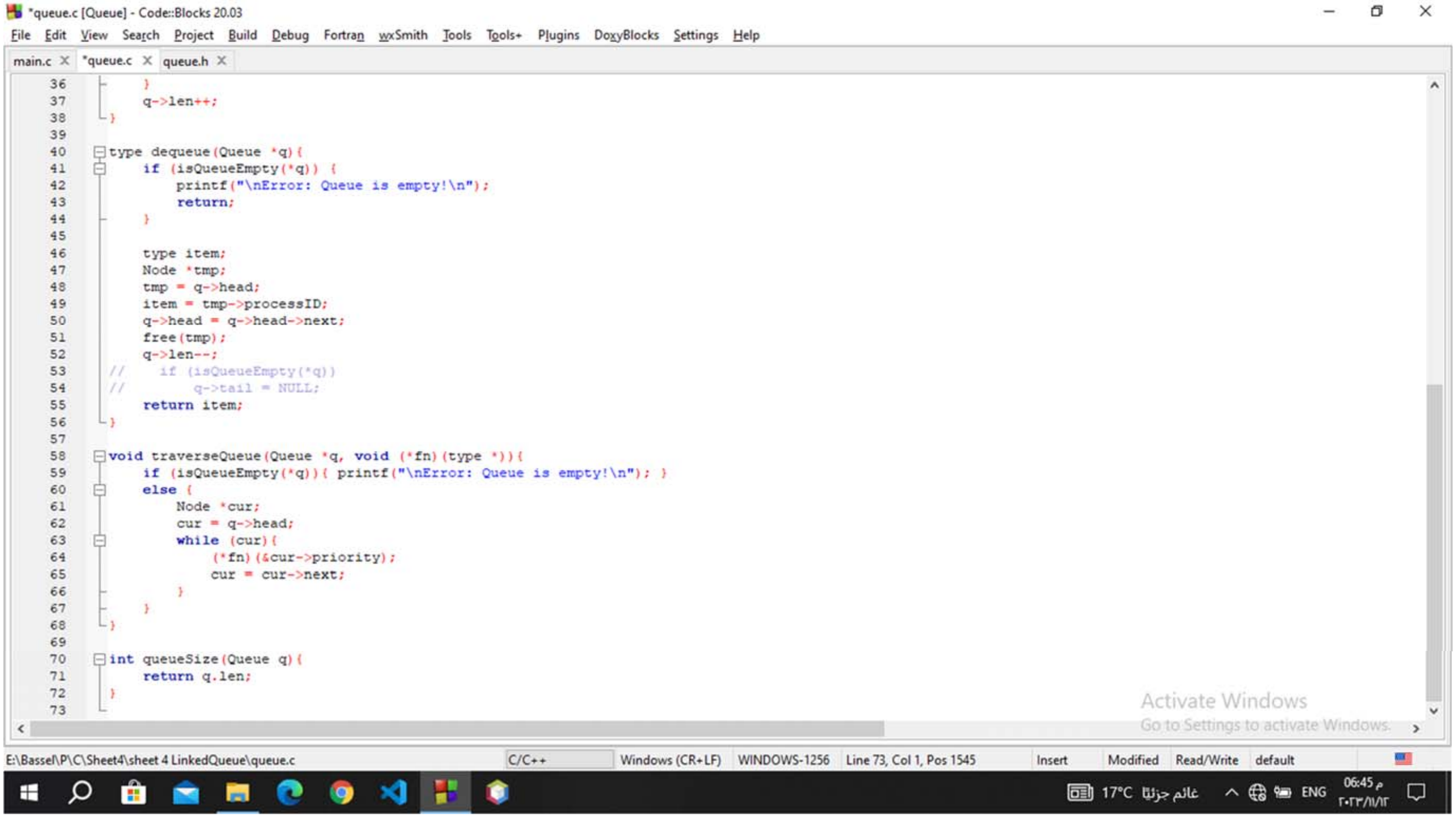

main.c [Queue] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.c queue.c queue.h
E:\Bassel\P\C\Sheet4\sheet 4 LinkedQueue\main.c
C/C++
Windows (CR+LF)
WINDO
"E:\Bassel\P\C\Sheet4\sheet 4 LinkedQueue\bin\Debug\Queue.exe"
Activate Windows
Go to Settings to activate Windows.
17°C غائم جزئيًا
ENG
06:44 م
٢٠٢٣/١١/١٢

```c
#include <stdio.h>
#include <stdlib.h>
#include "queue.h"

void printQueue(type *c){
    printf("%d\n", *c);
}

int main(){
    Queue Q;
    createQueue(&Q);

    int x, p_id, pri;
    printf("1. Add a New Process.\n"
           "2. Serve a Process.\n"
           "3. Number of Waiting Processes.\n"
           "4. Exit menu.\n");
    scanf("%d", &x);
    while (1){
        switch (x){
        case 1:
            printf("Enter the process ID: "); scanf("%d", &p_id);
            printf("Enter the priority of the process: "); scanf("%d", &pri);
            enqueue(&Q, p_id, pri);
            break;
        case 2:
            if (isQueueEmpty(Q)) dequeue(&Q);
            else printf("The process \"%d\" has been served successfully!\n", dequeue(&Q));
            break;
        case 3:
            printf("The number of waiting processes is -> %d\n", queueSize(Q));
            break;
        case 4:
            return 0;
        default:
            printf("Invalid choice!");
        }
        printf("\n1. Add a New Process.\n"
            "2. Serve a Process.\n"
            "3. Number of Waiting Processes.\n"
            "4. Exit menu.\n");
        scanf("%d", &x);
    }
```

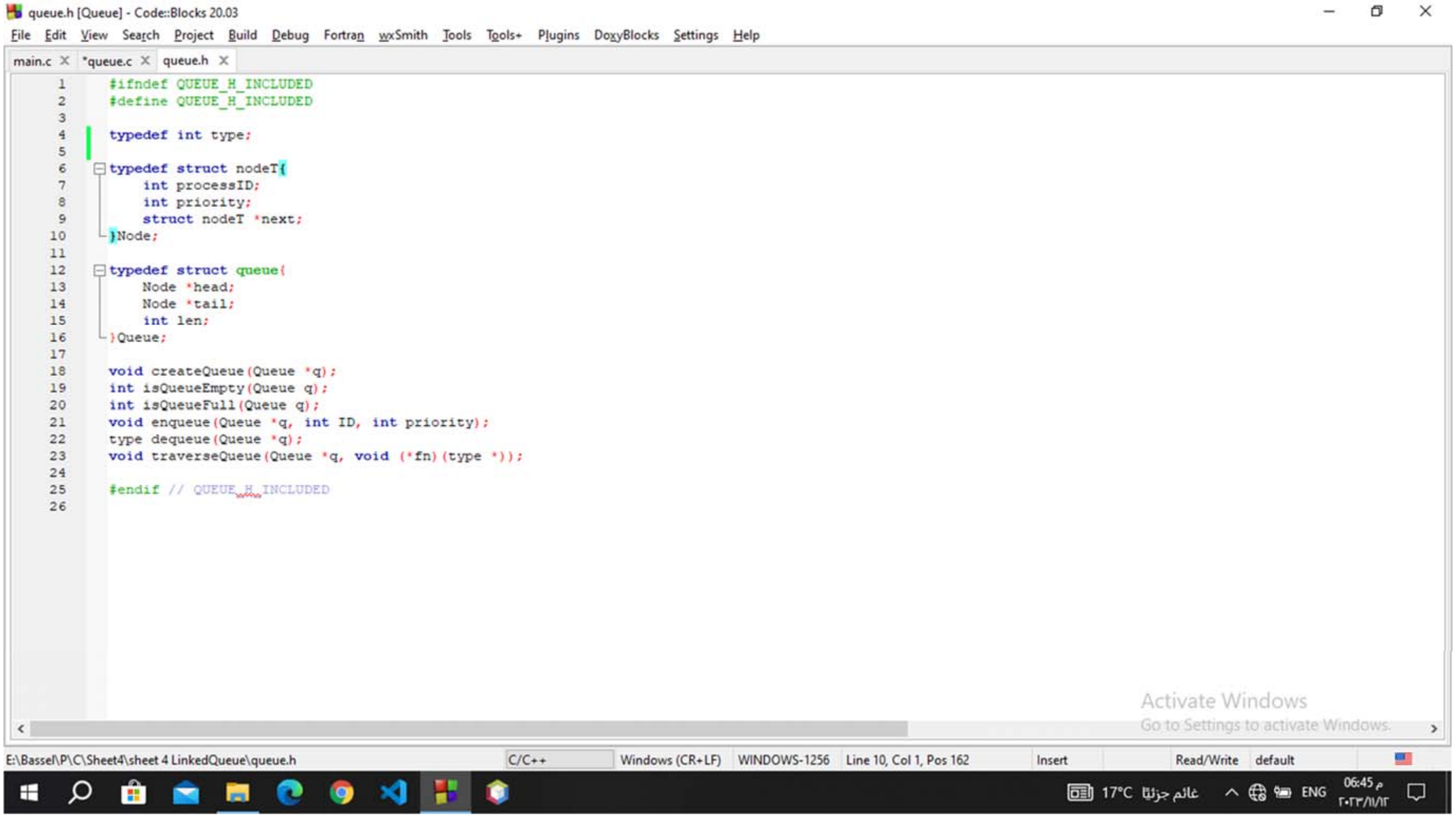
```
1. Add a New Process.
2. Serve a Process.
3. Number of Waiting Processes.
4. Exit menu.
1
Enter the process ID: 15
Enter the priority of the process: 2

1. Add a New Process.
2. Serve a Process.
3. Number of Waiting Processes.
4. Exit menu.
1
Enter the process ID: 64
Enter the priority of the process: 5

1. Add a New Process.
2. Serve a Process.
3. Number of Waiting Processes.
4. Exit menu.
16
Invalid choice!
1. Add a New Process.
2. Serve a Process.
3. Number of Waiting Processes.
4. Exit menu.
3
The number of waiting processes is -> 2

1. Add a New Process.
2. Serve a Process.
3. Number of Waiting Processes.
4. Exit menu.
2
The process "15" has been served successfully!

1. Add a New Process.
2. Serve a Process.
3. Number of Waiting Processes.
4. Exit menu.
2
The process "64" has been served successfully!

1. Add a New Process.
2. Serve a Process.
3. Number of Waiting Processes.
4. Exit menu.
2

Error: Queue is empty!

1. Add a New Process.
2. Serve a Process.
```


*queue.c [Queue] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.c
*queue.c
queue.h
Activate Windows
Go to Settings to activate Windows.
E:\Bassel\P\C\Sheet4\sheet 4 LinkedQueue\queue.c
C/C++
Windows (CR+LF)
WINDOWS-1256
Line 35, Col 23, Pos 836
Insert
Modified
Read/Write
default
17°C غائم جزئيًا
ENG
06:45 م
٢٠٢٣/١١/١٢

```c
#include <stdlib.h>
#include <stdio.h>
#include "queue.h"

void createQueue(Queue *q){
    q->head = NULL;
//    q->tail = NULL;
    q->len = 0;
}

int isQueueEmpty(Queue q){
    return (q.head==NULL);
}

int isQueueFull(Queue q){
    return 0;
}

void enqueue(Queue *q, int ID, int priority){
    Node *p = (Node *)malloc(sizeof(Node));
    p->processID = ID;
    p->priority = priority;
    if (isQueueEmpty(*q) || (p->priority<q->head->priority && p->priority!=q->head->priority)){
        p->next = q->head;
        q->head = p;
//        q->tail = p;
    } else {
        Node *tmp = (Node *)malloc(sizeof(Node));
        tmp = q->head;
        for (int i=0; i<q->len-1 && p->priority>tmp->priority; i++){
            tmp = tmp->next;
        }

        p->next = tmp->next;
        tmp->next = p;
    }
    q->len++;
}
```

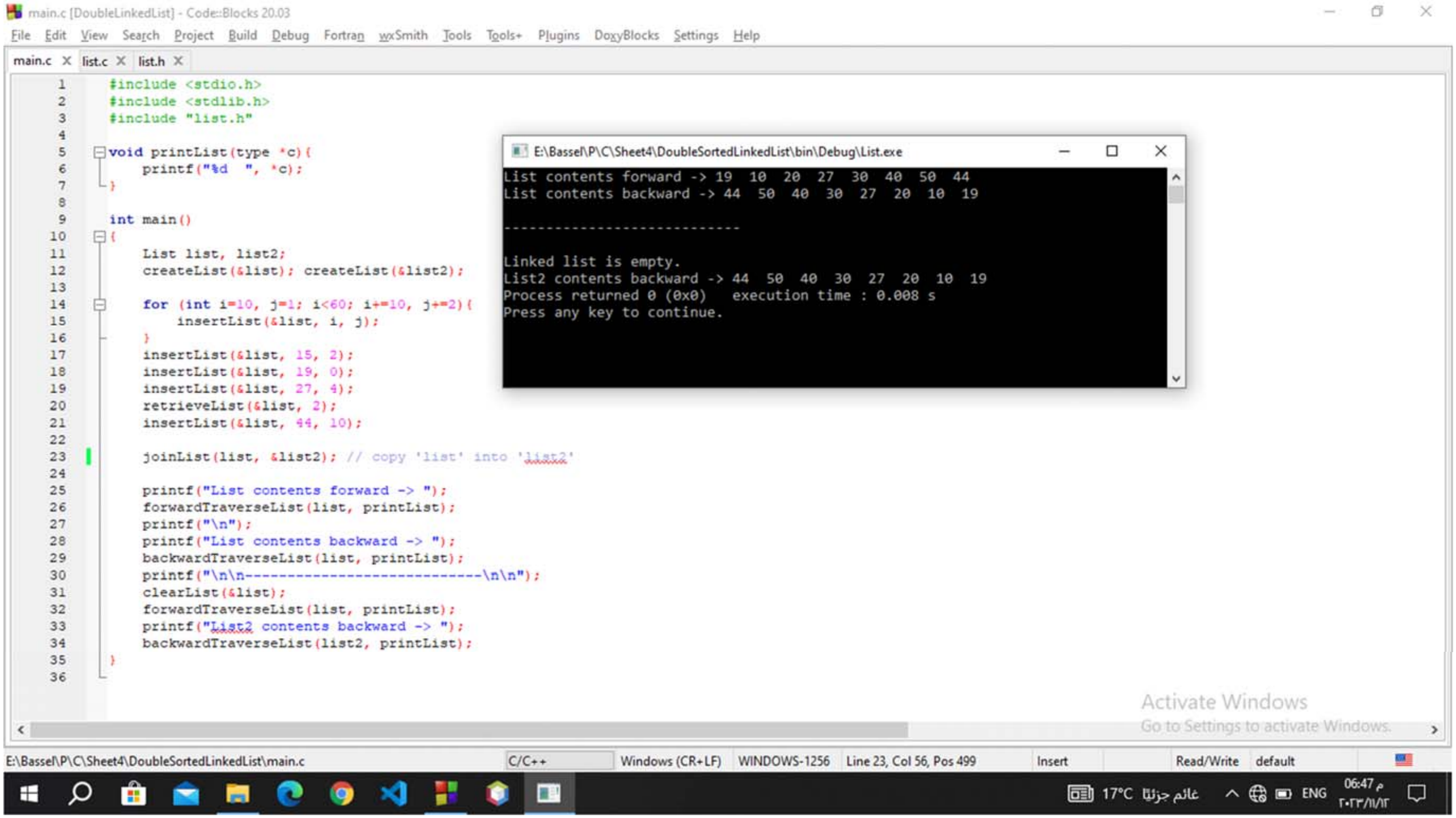


```c
    }
    q->len++;
}

type dequeue(Queue *q){
    if (isQueueEmpty(*q)) {
        printf("\nError: Queue is empty!\n");
        return;
    }

    type item;
    Node *tmp;
    tmp = q->head;
    item = tmp->processID;
    q->head = q->head->next;
    free(tmp);
    q->len--;
//      if (isQueueEmpty(*q))
//          q->tail = NULL;
    return item;
}

void traverseQueue(Queue *q, void (*fn)(type *)){
    if (isQueueEmpty(*q)){ printf("\nError: Queue is empty!\n"); }
    else {
        Node *cur;
        cur = q->head;
        while (cur){
            (*fn)(&cur->priority);
            cur = cur->next;
        }
    }
}

int queueSize(Queue q){
    return q.len;
}

```


queue.h [Queue] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.c
*queue.c
queue.h
Activate Windows
Go to Settings to activate Windows.
E:\Bassel\P\C\Sheet4\sheet 4 LinkedQueue\queue.h
C/C++
Windows (CR+LF)
WINDOWS-1256
Line 10, Col 1, Pos 162
Insert
Read/Write
default
17°C غائم جزئيا
ENG
06:45 م

```c
#ifndef QUEUE_H_INCLUDED
#define QUEUE_H_INCLUDED

typedef int type;

typedef struct nodeT{
    int processID;
    int priority;
    struct nodeT *next;
}Node;

typedef struct queue{
    Node *head;
    Node *tail;
    int len;
}Queue;

void createQueue(Queue *q);
int isQueueEmpty(Queue q);
int isQueueFull(Queue q);
void enqueue(Queue *q, int ID, int priority);
type dequeue(Queue *q);
void traverseQueue(Queue *q, void (*fn)(type *));

#endif // QUEUE_H_INCLUDED
```

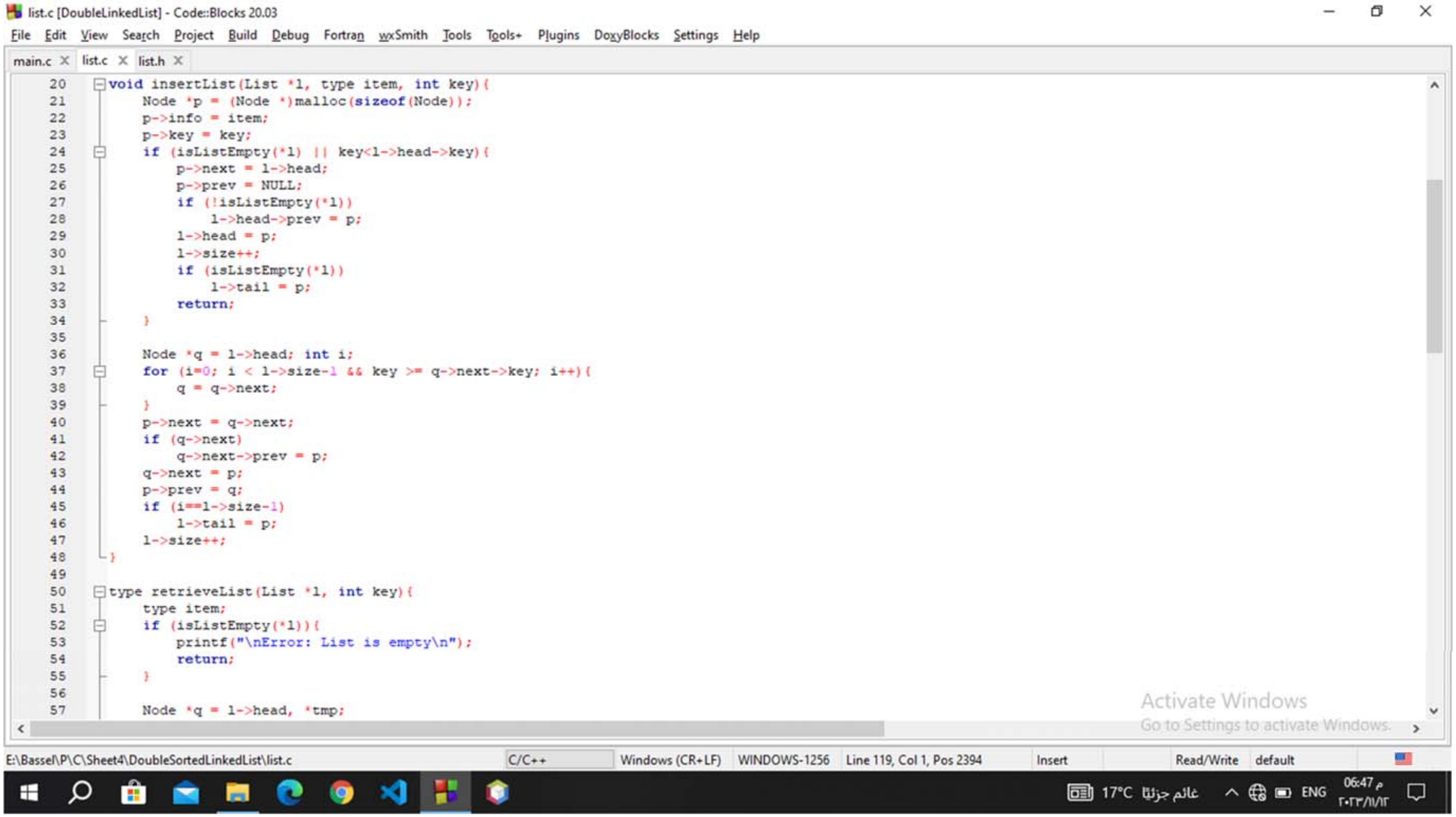
Q3:



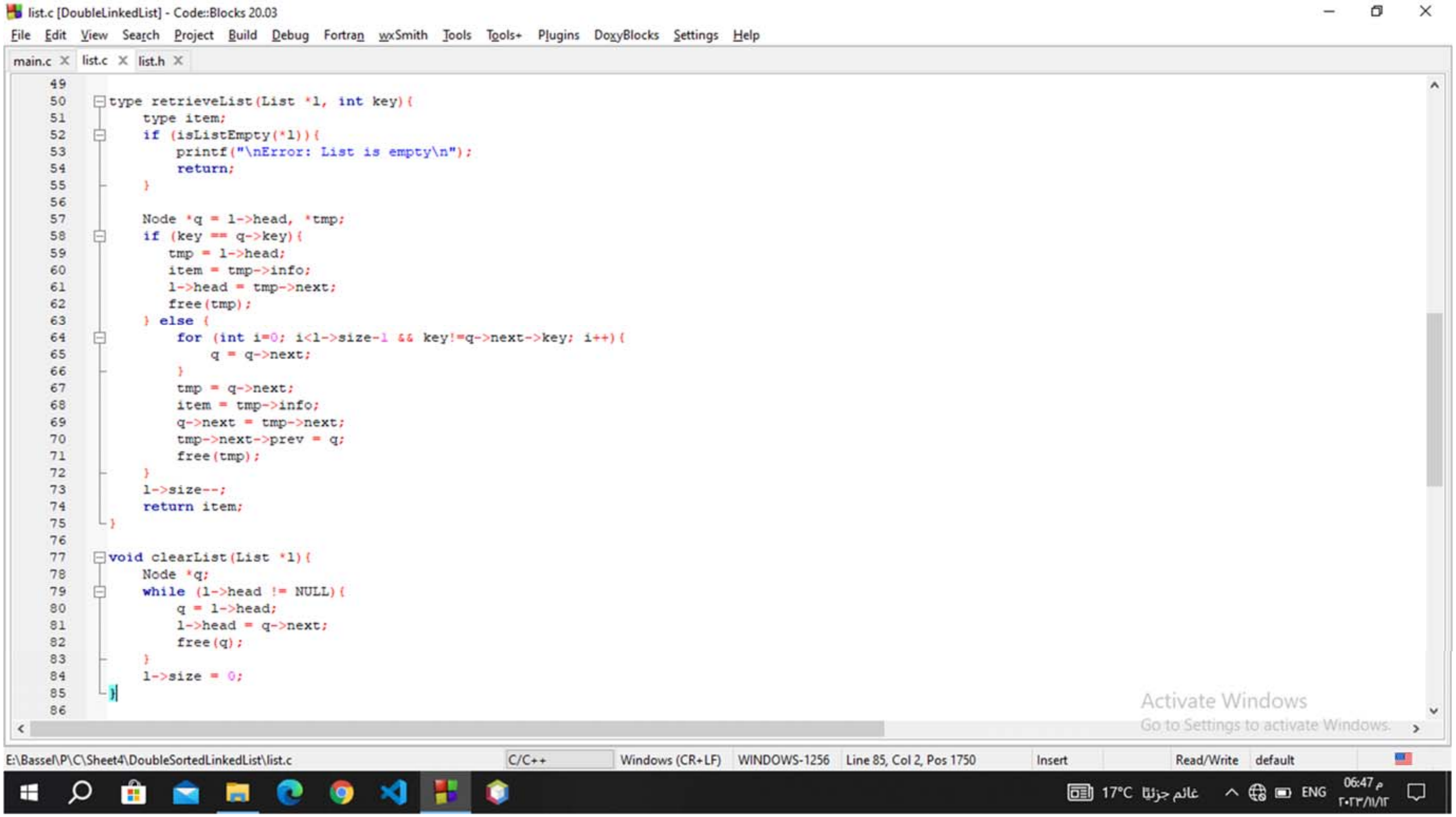
main.c [DoubleLinkedList] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.c list.c list.h
E:\Bassel\P\C\Sheet4\DoubleSortedLinkedList\bin\Debug\List.exe
Activate Windows
Go to Settings to activate Windows.
E:\Bassel\P\C\Sheet4\DoubleSortedLinkedList\main.c
C/C++
Windows (CR+LF)
WINDOWS-1256
Line 23, Col 56, Pos 499
Insert
Read/Write
default
17°C غائم جزئيًا
ENG
06:47 م
٢٠٢٣/١١/١٣

```c
#include <stdio.h>
#include <stdlib.h>
#include "list.h"

void printList(type *c){
    printf("%d  ", *c);
}

int main()
{
    List list, list2;
    createList(&list); createList(&list2);

    for (int i=10, j=1; i<60; i+=10, j+=2){
        insertList(&list, i, j);
    }
    insertList(&list, 15, 2);
    insertList(&list, 19, 0);
    insertList(&list, 27, 4);
    retrieveList(&list, 2);
    insertList(&list, 44, 10);

    joinList(list, &list2); // copy 'list' into 'list2'

    printf("List contents forward -> ");
    forwardTraverseList(list, printList);
    printf("\n");
    printf("List contents backward -> ");
    backwardTraverseList(list, printList);
    printf("\n\n---------------------------\n\n");
    clearList(&list);
    forwardTraverseList(list, printList);
    printf("List2 contents backward -> ");
    backwardTraverseList(list2, printList);
}

```

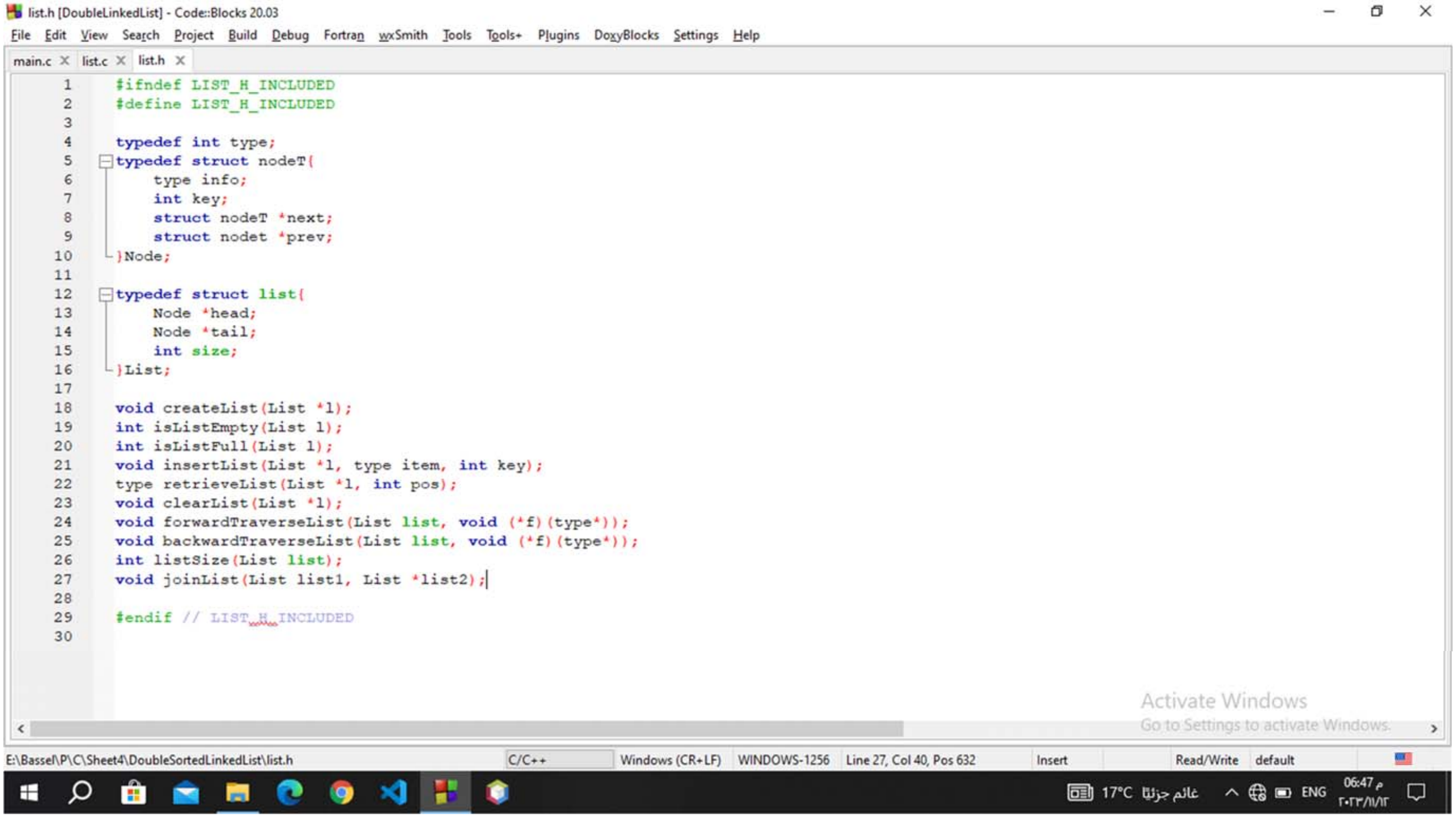
```
List contents forward -> 19  10  20  27  30  40  50  44
List contents backward -> 44  50  40  30  27  20  10  19

---------------------------

Linked list is empty.
List2 contents backward -> 44  50  40  30  27  20  10  19
Process returned 0 (0x0)   execution time : 0.008 s
Press any key to continue.
```



```c
void insertList(List *l, type item, int key){
    Node *p = (Node *)malloc(sizeof(Node));
    p->info = item;
    p->key = key;
    if (isListEmpty(*l) || key<l->head->key){
        p->next = l->head;
        p->prev = NULL;
        if (!isListEmpty(*l))
            l->head->prev = p;
        l->head = p;
        l->size++;
        if (isListEmpty(*l))
            l->tail = p;
        return;
    }

    Node *q = l->head; int i;
    for (i=0; i < l->size-1 && key >= q->next->key; i++){
        q = q->next;
    }
    p->next = q->next;
    if (q->next)
        q->next->prev = p;
    q->next = p;
    p->prev = q;
    if (i==l->size-1)
        l->tail = p;
    l->size++;
}

type retrieveList(List *l, int key){
    type item;
    if (isListEmpty(*l)){
        printf("\nError: List is empty\n");
        return;
    }

    Node *q = l->head, *tmp;
```


list.c [DoubleLinkedList] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.c
list.c
list.h
Activate Windows
Go to Settings to activate Windows.
E:\Bassel\P\C\Sheet4\DoubleSortedLinkedList\list.c
C/C++
Windows (CR+LF)
WINDOWS-1256
Line 85, Col 2, Pos 1750
Insert
Read/Write
default
17°C غائم جزئيا
ENG
06:47 م
٢٠٢٣/١١/١٢

```c

type retrieveList(List *l, int key){
    type item;
    if (isListEmpty(*l)){
        printf("\nError: List is empty\n");
        return;
    }

    Node *q = l->head, *tmp;
    if (key == q->key){
       tmp = l->head;
       item = tmp->info;
       l->head = tmp->next;
       free(tmp);
    } else {
        for (int i=0; i<l->size-1 && key!=q->next->key; i++){
            q = q->next;
        }
        tmp = q->next;
        item = tmp->info;
        q->next = tmp->next;
        tmp->next->prev = q;
        free(tmp);
    }
    l->size--;
    return item;
}

void clearList(List *l){
    Node *q;
    while (l->head != NULL){
        q = l->head;
        l->head = q->next;
        free(q);
    }
    l->size = 0;
}

```



```c
#ifndef LIST_H_INCLUDED
#define LIST_H_INCLUDED

typedef int type;
typedef struct nodeT{
    type info;
    int key;
    struct nodeT *next;
    struct nodet *prev;
}Node;

typedef struct list{
    Node *head;
    Node *tail;
    int size;
}List;

void createList(List *l);
int isListEmpty(List l);
int isListFull(List l);
void insertList(List *l, type item, int key);
type retrieveList(List *l, int pos);
void clearList(List *l);
void forwardTraverseList(List list, void (*f)(type*));
void backwardTraverseList(List list, void (*f)(type*));
int listSize(List list);
void joinList(List list1, List *list2);

#endif // LIST_H_INCLUDED

```