

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 X queue.c X queue.h X
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "queue.h"
4
5 void printQueue(type *c){
6     printf("%d\n", *c);
7 }
8
9 int main(){
10     Queue Q;
11     createQueue(&Q);
12
13     int x, p_id, pri;
14     printf("1. Add a New Process.\n"
15           "2. Serve a Process.\n"
16           "3. Number of Waiting Processes.\n"
17           "4. Exit menu.\n");
18     scanf("%d", &x);
19     while (1){
20         switch (x){
21             case 1:
22                 printf("Enter the process ID: "); scanf("%d", &p_id);
23                 printf("Enter the priority of the process: "); scanf("%d", &pri);
24                 enqueue(&Q, p_id, pri);
25                 break;
26             case 2:
27                 if (isEmpty(Q)) dequeue(&Q);
28                 else printf("The process \"%d\" has been served successfully!\n", dequeue(&Q));
29                 break;
30             case 3:
31                 printf("The number of waiting processes is -> %d\n", queueSize(Q));
32                 break;
33             case 4:
34                 return 0;
35             default:
36                 printf("Invalid choice!\n");
37         }
38         printf("\n1. Add a New Process.\n"
39               "2. Serve a Process.\n"
40               "3. Number of Waiting Processes.\n"
41               "4. Exit menu.\n");
42         scanf("%d", &x);
43     }
44 }

```

```

"E:\Bassef\PC\Sheet4\sheet 4 LinkedQueue\bin\Debug\Queue.exe"
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.

```

Activate Windows
Go to Settings to activate Windows.

E:\Bassef\PC\Sheet4\sheet 4 LinkedQueue\main.c C/C++ Windows (CR+LF) WINDOW

17°C عالم جزينا ENG 06:44 م ٢٠٢٢/١١/١٢

*queue.c [Queue] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

main.c X *queue.c X queue.h X

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include "queue.h"
4
5  void createQueue(Queue *q){
6      q->head = NULL;
7      // q->tail = NULL;
8      q->len = 0;
9  }
10
11 int isQueueEmpty(Queue q){
12     return (q.head==NULL);
13 }
14
15 int isQueueFull(Queue q){
16     return 0;
17 }
18
19 void enqueue(Queue *q, int ID, int priority){
20     Node *p = (Node *)malloc(sizeof(Node));
21     p->processID = ID;
22     p->priority = priority;
23     if (isQueueEmpty(*q) || (p->priority<q->head->priority && p->priority!=q->head->priority)){
24         p->next = q->head;
25         q->head = p;
26         // q->tail = p;
27     } else {
28         Node *tmp = (Node *)malloc(sizeof(Node));
29         tmp = q->head;
30         for (int i=0; i<q->len-1 && p->priority>tmp->priority; i++){
31             tmp = tmp->next;
32         }
33
34         p->next = tmp->next;
35         tmp->next = p;
36     }
37     q->len++;
38 }
```

Activate Windows
Go to Settings to activate Windows.

E:\Basse\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 عالم جزينا ENG 06:45 م ٢٠٢٢/١١/١٢

*queue.c [Queue] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

main.c X *queue.c X queue.h X

```
36     }
37     q->len++;
38 }
39
40 type dequeue(Queue *q){
41     if (isQueueEmpty(*q)) {
42         printf("\nError: Queue is empty!\n");
43         return;
44     }
45
46     type item;
47     Node *tmp;
48     tmp = q->head;
49     item = tmp->processID;
50     q->head = q->head->next;
51     free(tmp);
52     q->len--;
53     // if (isQueueEmpty(*q))
54     // q->tail = NULL;
55     return item;
56 }
57
58 void traverseQueue(Queue *q, void (*fn)(type *)){
59     if (isQueueEmpty(*q)) { printf("\nError: Queue is empty!\n"); }
60     else {
61         Node *cur;
62         cur = q->head;
63         while (cur){
64             (*fn)(&cur->priority);
65             cur = cur->next;
66         }
67     }
68 }
69
70 int queueSize(Queue q){
71     return q.len;
72 }
73 }
```

Activate Windows
Go to Settings to activate Windows.

E:\Basse\p\c\Sheet4\sheet 4 LinkedQueue\queue.c C/C++ Windows (CR+LF) WINDOWS-1256 Line 73, Col 1, Pos 1545 Insert Modified Read/Write default

17°C عالم جزنبا ENG 06:45 م ٢٠٢٣/١١/١٢

queue.h [Queue] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

main.c x *queue.c x queue.h x

```
1  #ifndef QUEUE_H_INCLUDED
2  #define QUEUE_H_INCLUDED
3
4  typedef int type;
5
6  typedef struct nodeT{
7      int processID;
8      int priority;
9      struct nodeT *next;
10 }Node;
11
12 typedef struct queue{
13     Node *head;
14     Node *tail;
15     int len;
16 }Queue;
17
18 void createQueue(Queue *q);
19 int isEmptyQueue(Queue q);
20 int isQueueFull(Queue q);
21 void enqueue(Queue *q, int ID, int priority);
22 type dequeue(Queue *q);
23 void traverseQueue(Queue *q, void (*fn)(type *));
24
25 #endif // QUEUE_H_INCLUDED
26
```

Activate Windows
Go to Settings to activate Windows.

E:\Basse\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 عالم جزينا ENG 06:45 م ٢٠٢٢/١١/١٢

Q3:

The image shows a C++ IDE window titled "main.c [DoubleLinkedList] - Code::Blocks 20.03". The code in "main.c" defines a linked list structure and implements functions for creating, inserting, traversing, and clearing the list. The main function creates two lists, 'list' and 'list2', and demonstrates various operations. A terminal window titled "E:\Bassel\p\C\Sheet4\DoubleSortedLinkedList\bin\Debug\List.exe" shows the output of the program, which prints the contents of the lists forward and backward, followed by a message indicating the linked list is empty and the execution time.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "list.h"
4
5 void printList(type *c){
6     printf("%d ", *c);
7 }
8
9 int main()
10 {
11     List list, list2;
12     createList(&list); createList(&list2);
13
14     for (int i=10, j=1; i<60; i+=10, j+=2){
15         insertList(&list, i, j);
16     }
17     insertList(&list, 15, 2);
18     insertList(&list, 19, 0);
19     insertList(&list, 27, 4);
20     retrieveList(&list, 2);
21     insertList(&list, 44, 10);
22
23     joinList(list, &list2); // copy 'list' into 'list2'
24
25     printf("List contents forward -> ");
26     forwardTraverseList(list, printList);
27     printf("\n");
28     printf("List contents backward -> ");
29     backwardTraverseList(list, printList);
30     printf("\n\n-----\n\n");
31     clearList(&list);
32     forwardTraverseList(list, printList);
33     printf("List2 contents backward -> ");
34     backwardTraverseList(list2, printList);
35 }
36
```

```
E:\Bassel\p\C\Sheet4\DoubleSortedLinkedList\bin\Debug\List.exe
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.
```

```

20 void insertList(List *l, type item, int key){
21     Node *p = (Node *)malloc(sizeof(Node));
22     p->info = item;
23     p->key = key;
24     if (isListEmpty(*l) || key < l->head->key){
25         p->next = l->head;
26         p->prev = NULL;
27         if (!isListEmpty(*l))
28             l->head->prev = p;
29         l->head = p;
30         l->size++;
31         if (isListEmpty(*l))
32             l->tail = p;
33         return;
34     }
35
36     Node *q = l->head; int i;
37     for (i=0; i < l->size-1 && key >= q->next->key; i++){
38         q = q->next;
39     }
40     p->next = q->next;
41     if (q->next)
42         q->next->prev = p;
43     q->next = p;
44     p->prev = q;
45     if (i==l->size-1)
46         l->tail = p;
47     l->size++;
48 }
49
50 type retrieveList(List *l, int key){
51     type item;
52     if (isListEmpty(*l)){
53         printf("\nError: List is empty\n");
54         return;
55     }
56
57     Node *q = l->head, *tmp;

```

Activate Windows
Go to Settings to activate Windows.

list.c [DoubleLinkedList] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

main.c x list.c x list.h x

```
49
50 type retrieveList(List *l, int key){
51     type item;
52     if (isEmpty(*l)){
53         printf("\nError: List is empty\n");
54         return;
55     }
56
57     Node *q = l->head, *tmp;
58     if (key == q->key){
59         tmp = l->head;
60         item = tmp->info;
61         l->head = tmp->next;
62         free(tmp);
63     } else {
64         for (int i=0; i<l->size-1 && key!=q->next->key; i++){
65             q = q->next;
66         }
67         tmp = q->next;
68         item = tmp->info;
69         q->next = tmp->next;
70         tmp->next->prev = q;
71         free(tmp);
72     }
73     l->size--;
74     return item;
75 }
76
77 void clearList(List *l){
78     Node *q;
79     while (l->head != NULL){
80         q = l->head;
81         l->head = q->next;
82         free(q);
83     }
84     l->size = 0;
85 }
86 }
```

Activate Windows
Go to Settings to activate Windows.

E:\Basse\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 عالم جزئیة ENG 06:47 م ٢٠٢٢/١١/١٢

list.h [DoubleLinkedList] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

main.c x list.c x list.h x

```
1  #ifndef LIST_H_INCLUDED
2  #define LIST_H_INCLUDED
3
4  typedef int type;
5  typedef struct nodeT{
6      type info;
7      int key;
8      struct nodeT *next;
9      struct nodeT *prev;
10 }Node;
11
12 typedef struct list{
13     Node *head;
14     Node *tail;
15     int size;
16 }List;
17
18 void createList(List *l);
19 int isListEmpty(List l);
20 int isListFull(List l);
21 void insertList(List *l, type item, int key);
22 type retrieveList(List *l, int pos);
23 void clearList(List *l);
24 void forwardTraverseList(List list, void (*f)(type*));
25 void backwardTraverseList(List list, void (*f)(type*));
26 int listSize(List list);
27 void joinList(List list1, List *list2);
28
29 #endif // LIST_H_INCLUDED
30
```

Activate Windows
Go to Settings to activate Windows.

E:\Basse\p\c\Sheet4\DoubleSortedLinkedList\list.h C/C++ Windows (CR+LF) WINDOWS-1256 Line 27, Col 40, Pos 632 Insert Read/Write default

17°C عالم جزئیة ENG 06:47 م ٢٠٢٢/١١/١٢