TP MODUL 8

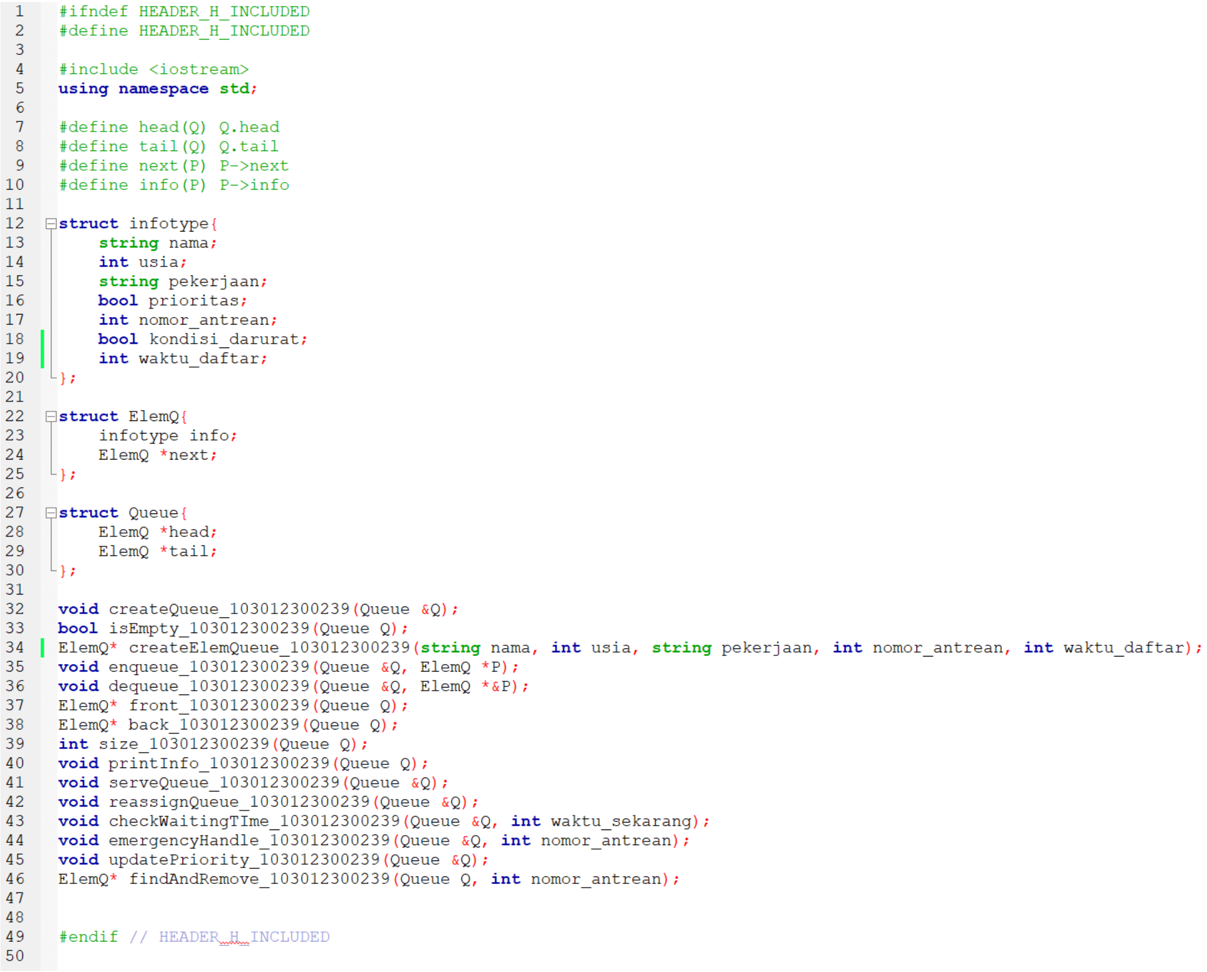
Nama : **Ahmad Naufal Ramadhan**

NIM : **103012300239**

Kelas : **IF-47-02**

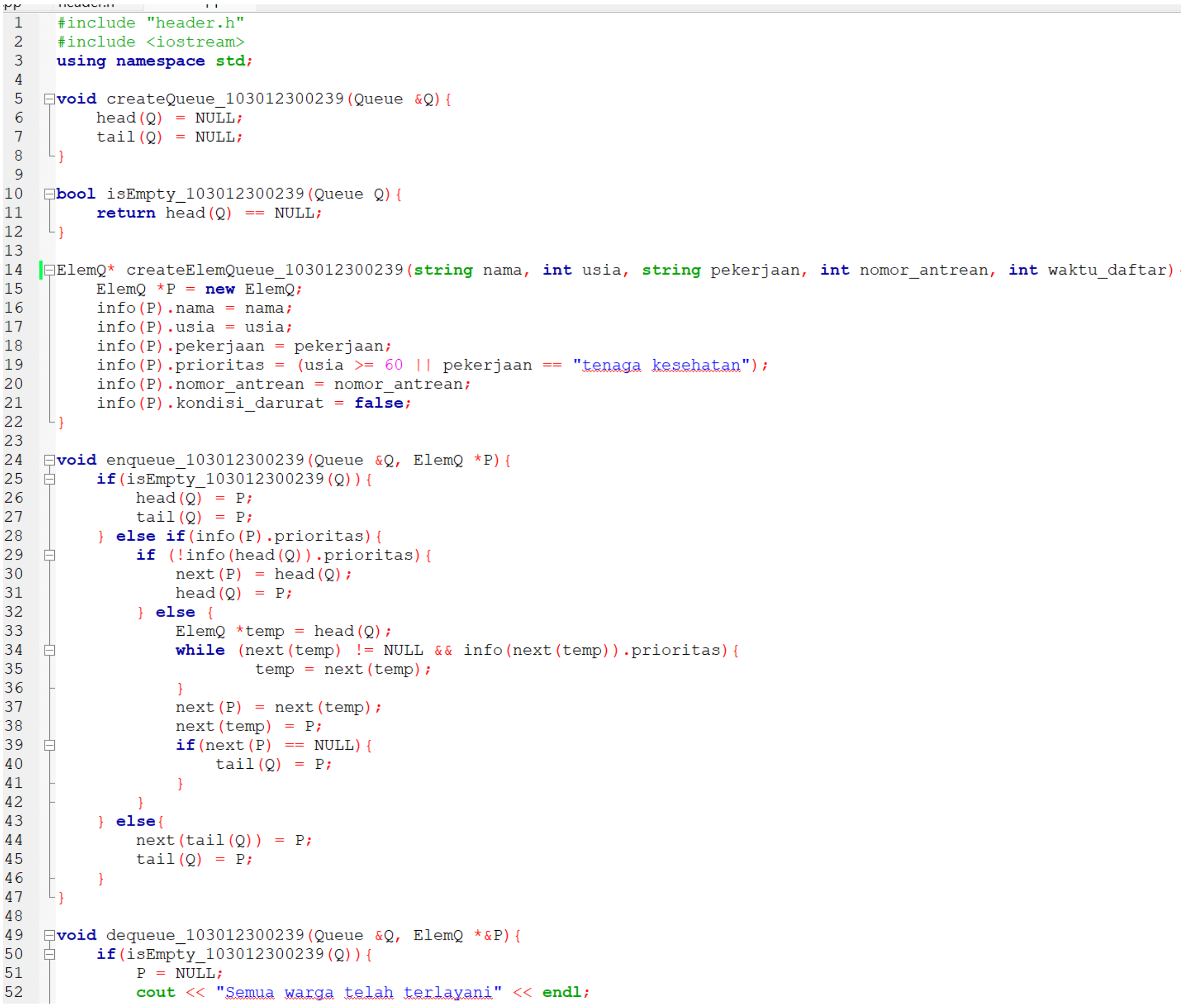
Kode Asprak : **YDA**

Header.h

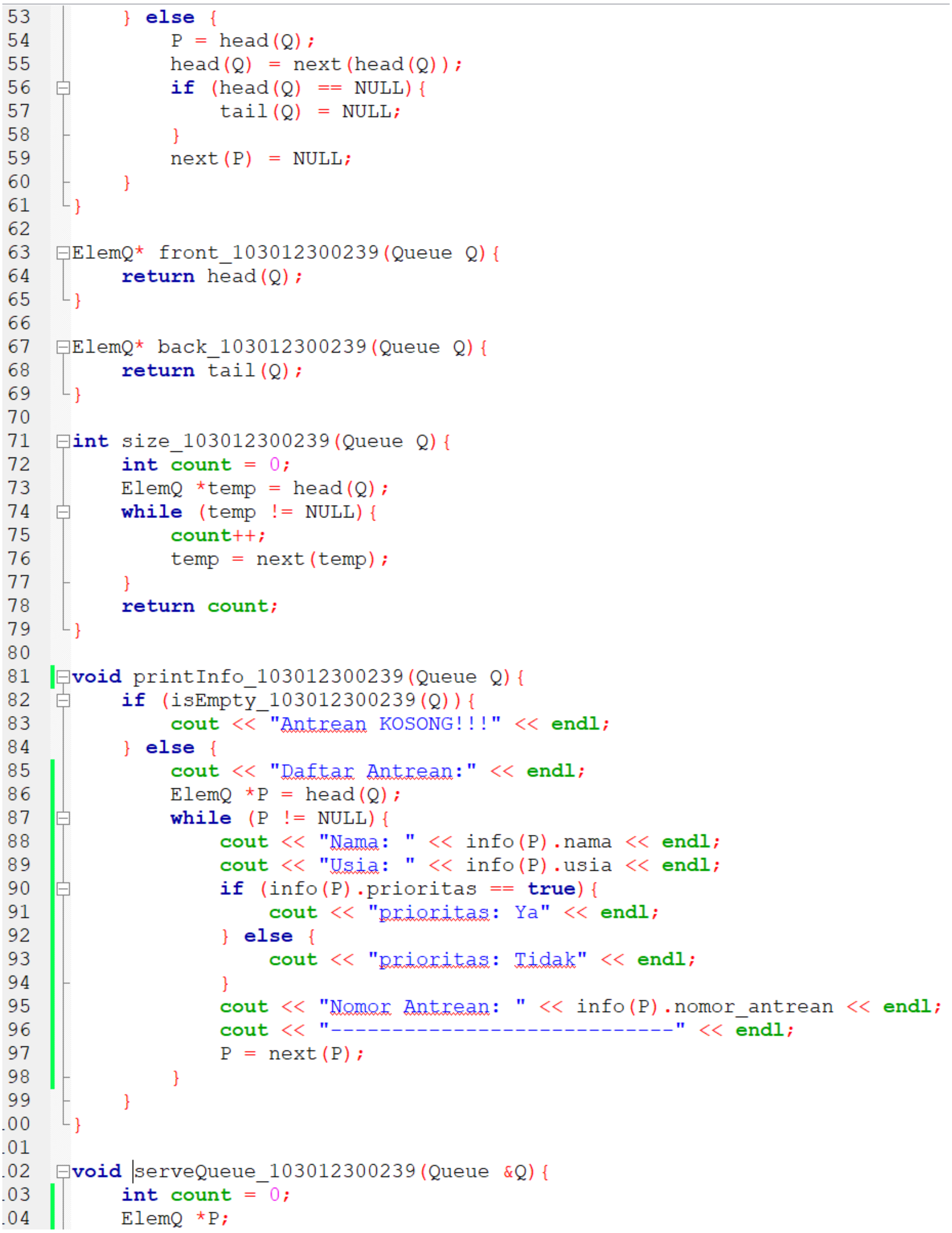


```cpp
#ifndef HEADER_H_INCLUDED
#define HEADER_H_INCLUDED

#include <iostream>
using namespace std;

#define head(Q) Q.head
#define tail(Q) Q.tail
#define next(P) P->next
#define info(P) P->info

struct infotype{
    string nama;
    int usia;
    string pekerjaan;
    bool prioritas;
    int nomor_antrean;
    bool kondisi_darurat;
    int waktu_daftar;
};

struct ElemQ{
    infotype info;
    ElemQ *next;
};

struct Queue{
    ElemQ *head;
    ElemQ *tail;
};

void createQueue_103012300239(Queue &Q);
bool isEmpty_103012300239(Queue Q);
ElemQ* createElemQueue_103012300239(string nama, int usia, string pekerjaan, int nomor_antrean, int waktu_daftar);
void enqueue_103012300239(Queue &Q, ElemQ *P);
void dequeue_103012300239(Queue &Q, ElemQ *&P);
ElemQ* front_103012300239(Queue Q);
ElemQ* back_103012300239(Queue Q);
int size_103012300239(Queue Q);
void printInfo_103012300239(Queue Q);
void serveQueue_103012300239(Queue &Q);
void reassignQueue_103012300239(Queue &Q);
void checkWaitingTIme_103012300239(Queue &Q, int waktu_sekarang);
void emergencyHandle_103012300239(Queue &Q, int nomor_antrean);
void updatePriority_103012300239(Queue &Q);
ElemQ* findAndRemove_103012300239(Queue Q, int nomor_antrean);


#endif // HEADER_H_INCLUDED

```

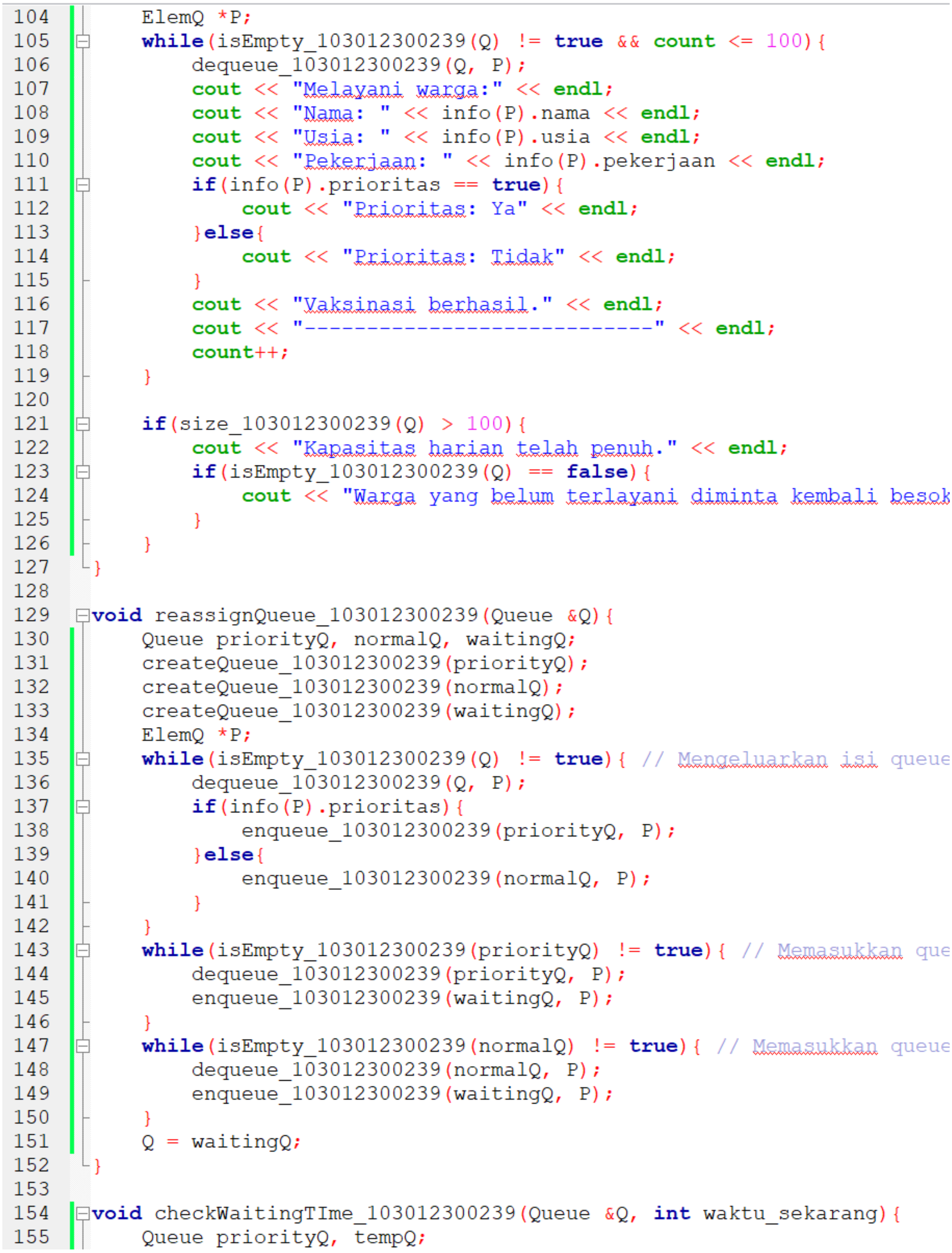
Header.cpp



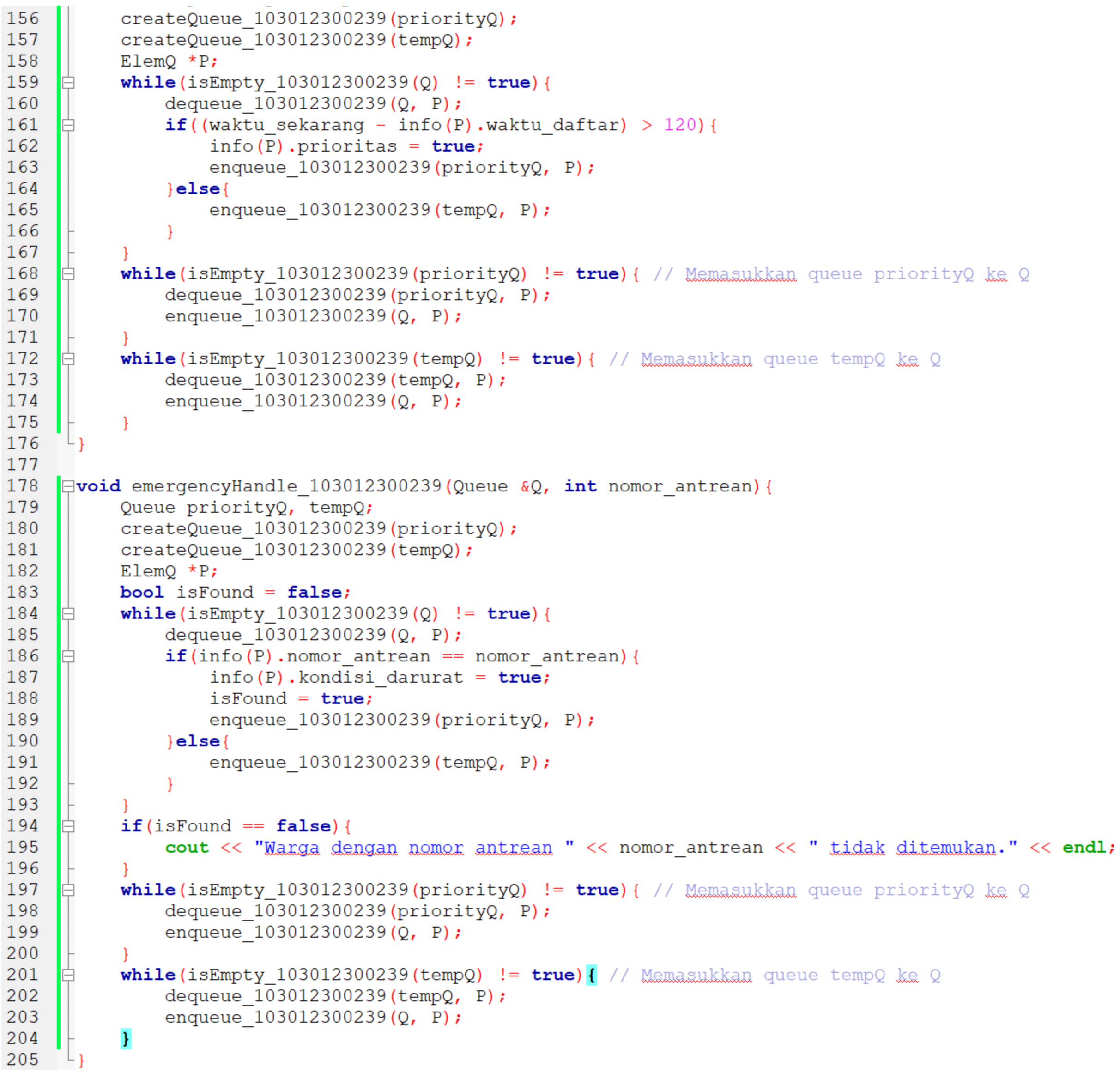
```cpp
#include "header.h"
#include <iostream>
using namespace std;

void createQueue_103012300239(Queue &Q){
    head(Q) = NULL;
    tail(Q) = NULL;
}

bool isEmpty_103012300239(Queue Q){
    return head(Q) == NULL;
}

ElemQ* createElemQueue_103012300239(string nama, int usia, string pekerjaan, int nomor_antrean, int waktu_daftar)
    ElemQ *P = new ElemQ;
    info(P).nama = nama;
    info(P).usia = usia;
    info(P).pekerjaan = pekerjaan;
    info(P).prioritas = (usia >= 60 || pekerjaan == "tenaga kesehatan");
    info(P).nomor_antrean = nomor_antrean;
    info(P).kondisi_darurat = false;
}

void enqueue_103012300239(Queue &Q, ElemQ *P){
    if(isEmpty_103012300239(Q)){
        head(Q) = P;
        tail(Q) = P;
    } else if(info(P).prioritas){
        if (!info(head(Q)).prioritas){
            next(P) = head(Q);
            head(Q) = P;
        } else {
            ElemQ *temp = head(Q);
            while (next(temp) != NULL && info(next(temp)).prioritas){
                    temp = next(temp);
            }
            next(P) = next(temp);
            next(temp) = P;
            if(next(P) == NULL){
                tail(Q) = P;
            }
        }
    } else{
        next(tail(Q)) = P;
        tail(Q) = P;
    }
}

void dequeue_103012300239(Queue &Q, ElemQ *&P){
    if(isEmpty_103012300239(Q)){
        P = NULL;
        cout << "Semua warga telah terlayani" << endl;
```



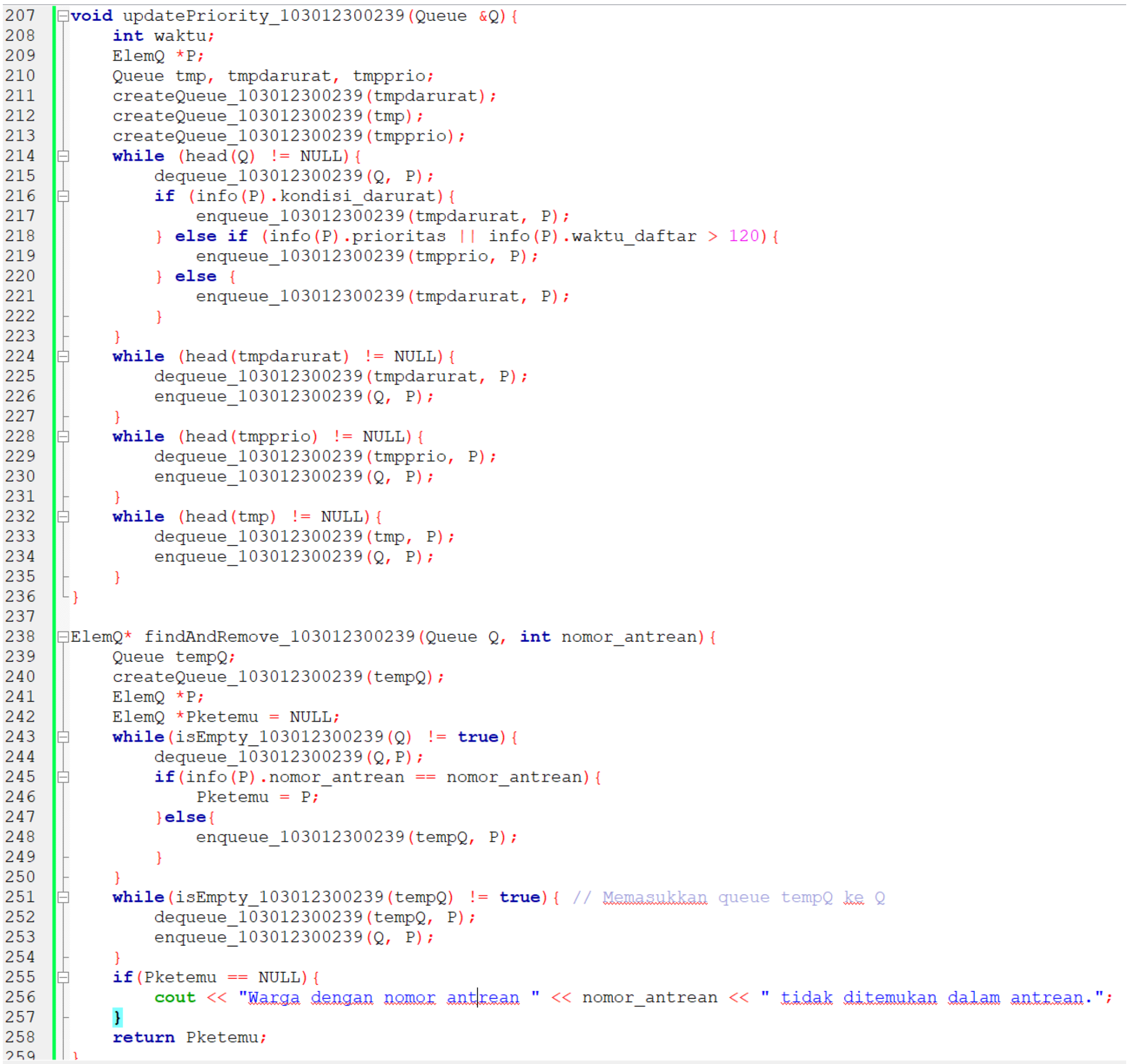
```
    } else {
        P = head(Q);
        head(Q) = next(head(Q));
        if (head(Q) == NULL){
            tail(Q) = NULL;
        }
        next(P) = NULL;
    }
}

ElemQ* front_103012300239(Queue Q){
    return head(Q);
}

ElemQ* back_103012300239(Queue Q){
    return tail(Q);
}

int size_103012300239(Queue Q){
    int count = 0;
    ElemQ *temp = head(Q);
    while (temp != NULL){
        count++;
        temp = next(temp);
    }
    return count;
}

void printInfo_103012300239(Queue Q){
    if (isEmpty_103012300239(Q)){
        cout << "Antrean KOSONG!!!" << endl;
    } else {
        cout << "Daftar Antrean:" << endl;
        ElemQ *P = head(Q);
        while (P != NULL){
            cout << "Nama: " << info(P).nama << endl;
            cout << "Usia: " << info(P).usia << endl;
            if (info(P).prioritas == true){
                cout << "prioritas: Ya" << endl;
            } else {
                cout << "prioritas: Tidak" << endl;
            }
            cout << "Nomor Antrean: " << info(P).nomor_antrean << endl;
            cout << "----------------------------" << endl;
            P = next(P);
        }
    }
}

void serveQueue_103012300239(Queue &Q){
    int count = 0;
    ElemQ *P;
```



```
    ElemQ *P;
    while(isEmpty_103012300239(Q) != true && count <= 100){
        dequeue_103012300239(Q, P);
        cout << "Melayani warga:" << endl;
        cout << "Nama: " << info(P).nama << endl;
        cout << "Usia: " << info(P).usia << endl;
        cout << "Pekerjaan: " << info(P).pekerjaan << endl;
        if(info(P).prioritas == true){
            cout << "Prioritas: Ya" << endl;
        }else{
            cout << "Prioritas: Tidak" << endl;
        }
        cout << "Vaksinasi berhasil." << endl;
        cout << "----------------------------" << endl;
        count++;
    }

    if(size_103012300239(Q) > 100){
        cout << "Kapasitas harian telah penuh." << endl;
        if(isEmpty_103012300239(Q) == false){
            cout << "Warga yang belum terlayani diminta kembali besok
        }
    }
}

void reassignQueue_103012300239(Queue &Q){
    Queue priorityQ, normalQ, waitingQ;
    createQueue_103012300239(priorityQ);
    createQueue_103012300239(normalQ);
    createQueue_103012300239(waitingQ);
    ElemQ *P;
    while(isEmpty_103012300239(Q) != true){ // Mengeluarkan isi queue
        dequeue_103012300239(Q, P);
        if(info(P).prioritas){
            enqueue_103012300239(priorityQ, P);
        }else{
            enqueue_103012300239(normalQ, P);
        }
    }
    while(isEmpty_103012300239(priorityQ) != true){ // Memasukkan que
        dequeue_103012300239(priorityQ, P);
        enqueue_103012300239(waitingQ, P);
    }
    while(isEmpty_103012300239(normalQ) != true){ // Memasukkan queue
        dequeue_103012300239(normalQ, P);
        enqueue_103012300239(waitingQ, P);
    }
    Q = waitingQ;
}

void checkWaitingTIme_103012300239(Queue &Q, int waktu_sekarang){
    Queue priorityQ, tempQ;
```

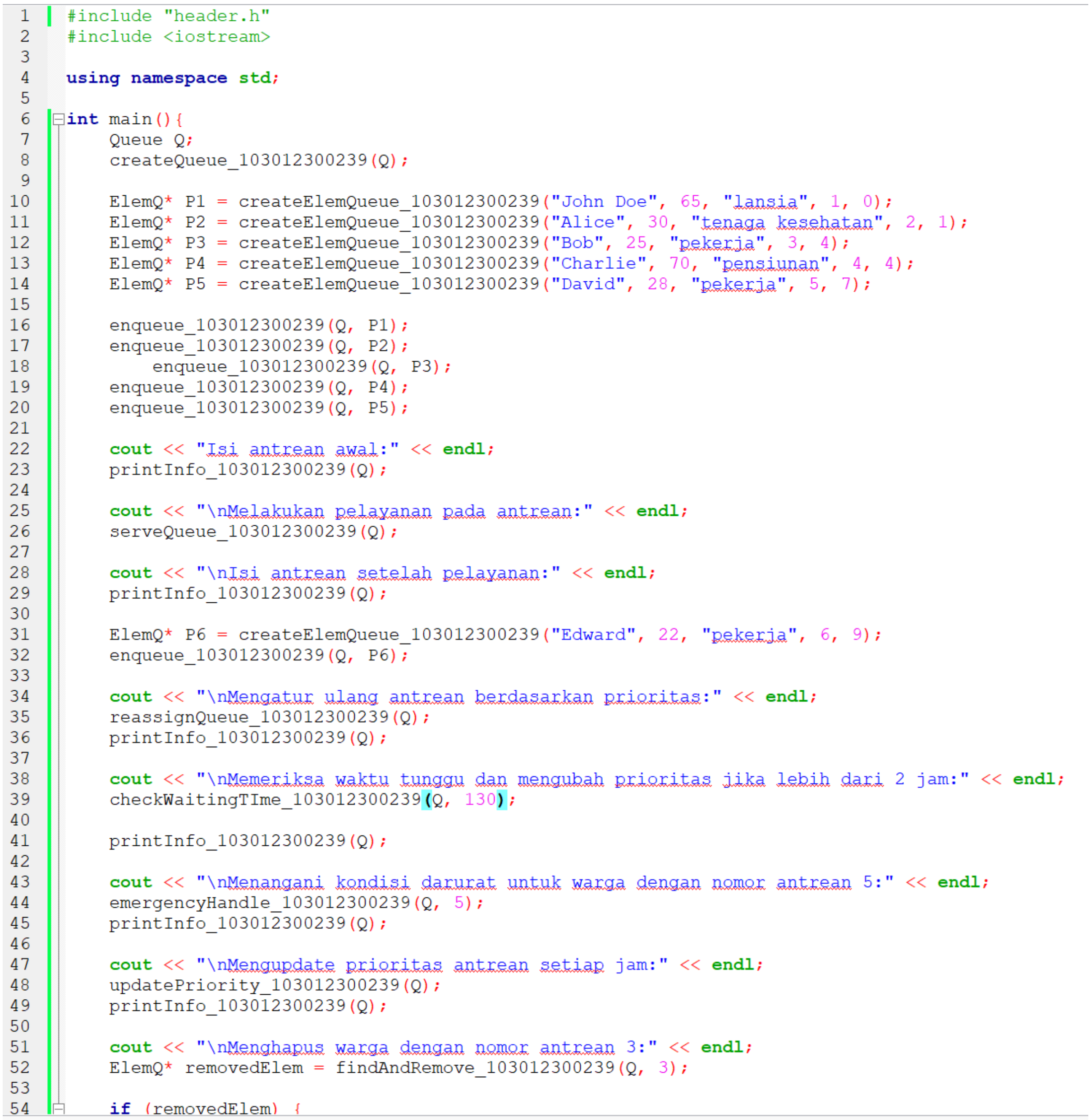


```
    createQueue_103012300239(priorityQ);
    createQueue_103012300239(tempQ);
    ElemQ *P;
    while(isEmpty_103012300239(Q) != true){
        dequeue_103012300239(Q, P);
        if((waktu_sekarang - info(P).waktu_daftar) > 120){
            info(P).prioritas = true;
            enqueue_103012300239(priorityQ, P);
        }else{
            enqueue_103012300239(tempQ, P);
        }
    }
    while(isEmpty_103012300239(priorityQ) != true){ // Memasukkan queue priorityQ ke Q
        dequeue_103012300239(priorityQ, P);
        enqueue_103012300239(Q, P);
    }
    while(isEmpty_103012300239(tempQ) != true){ // Memasukkan queue tempQ ke Q
        dequeue_103012300239(tempQ, P);
        enqueue_103012300239(Q, P);
    }
}

void emergencyHandle_103012300239(Queue &Q, int nomor_antrean){
    Queue priorityQ, tempQ;
    createQueue_103012300239(priorityQ);
    createQueue_103012300239(tempQ);
    ElemQ *P;
    bool isFound = false;
    while(isEmpty_103012300239(Q) != true){
        dequeue_103012300239(Q, P);
        if(info(P).nomor_antrean == nomor_antrean){
            info(P).kondisi_darurat = true;
            isFound = true;
            enqueue_103012300239(priorityQ, P);
        }else{
            enqueue_103012300239(tempQ, P);
        }
    }
    if(isFound == false){
        cout << "Warga dengan nomor antrean " << nomor_antrean << " tidak ditemukan." << endl;
    }
    while(isEmpty_103012300239(priorityQ) != true){ // Memasukkan queue priorityQ ke Q
        dequeue_103012300239(priorityQ, P);
        enqueue_103012300239(Q, P);
    }
    while(isEmpty_103012300239(tempQ) != true){ // Memasukkan queue tempQ ke Q
        dequeue_103012300239(tempQ, P);
        enqueue_103012300239(Q, P);
    }
}
```

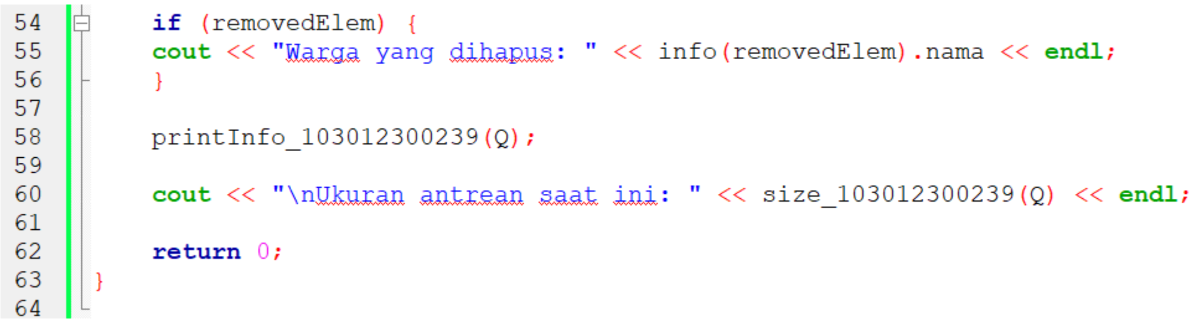


```cpp
void updatePriority_103012300239(Queue &Q){
    int waktu;
    ElemQ *P;
    Queue tmp, tmpdarurat, tmpprio;
    createQueue_103012300239(tmpdarurat);
    createQueue_103012300239(tmp);
    createQueue_103012300239(tmpprio);
    while (head(Q) != NULL){
        dequeue_103012300239(Q, P);
        if (info(P).kondisi_darurat){
            enqueue_103012300239(tmpdarurat, P);
        } else if (info(P).prioritas || info(P).waktu_daftar > 120){
            enqueue_103012300239(tmpprio, P);
        } else {
            enqueue_103012300239(tmpdarurat, P);
        }
    }
    while (head(tmpdarurat) != NULL){
        dequeue_103012300239(tmpdarurat, P);
        enqueue_103012300239(Q, P);
    }
    while (head(tmpprio) != NULL){
        dequeue_103012300239(tmpprio, P);
        enqueue_103012300239(Q, P);
    }
    while (head(tmp) != NULL){
        dequeue_103012300239(tmp, P);
        enqueue_103012300239(Q, P);
    }
}

ElemQ* findAndRemove_103012300239(Queue Q, int nomor_antrean){
    Queue tempQ;
    createQueue_103012300239(tempQ);
    ElemQ *P;
    ElemQ *Pketemu = NULL;
    while(isEmpty_103012300239(Q) != true){
        dequeue_103012300239(Q,P);
        if(info(P).nomor_antrean == nomor_antrean){
            Pketemu = P;
        }else{
            enqueue_103012300239(tempQ, P);
        }
    }
    while(isEmpty_103012300239(tempQ) != true){ // Memasukkan queue tempQ ke Q
        dequeue_103012300239(tempQ, P);
        enqueue_103012300239(Q, P);
    }
    if(Pketemu == NULL){
        cout << "Warga dengan nomor antrean " << nomor_antrean << " tidak ditemukan dalam antrean.";
    }
    return Pketemu;
}
```

Main.cpp

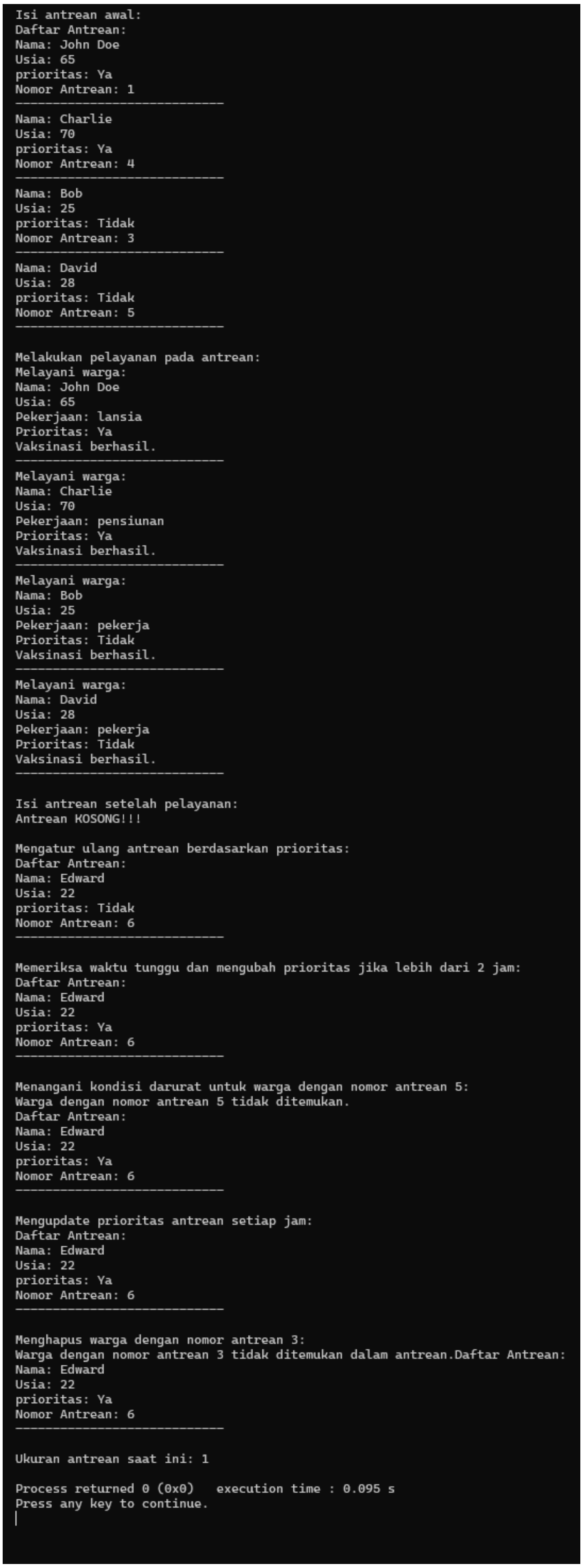


```cpp
#include "header.h"
#include <iostream>

using namespace std;

int main(){
    Queue Q;
    createQueue_103012300239(Q);

    ElemQ* P1 = createElemQueue_103012300239("John Doe", 65, "lansia", 1, 0);
    ElemQ* P2 = createElemQueue_103012300239("Alice", 30, "tenaga kesehatan", 2, 1);
    ElemQ* P3 = createElemQueue_103012300239("Bob", 25, "pekerja", 3, 4);
    ElemQ* P4 = createElemQueue_103012300239("Charlie", 70, "pensiunan", 4, 4);
    ElemQ* P5 = createElemQueue_103012300239("David", 28, "pekerja", 5, 7);

    enqueue_103012300239(Q, P1);
    enqueue_103012300239(Q, P2);
        enqueue_103012300239(Q, P3);
    enqueue_103012300239(Q, P4);
    enqueue_103012300239(Q, P5);

    cout << "Isi antrean awal:" << endl;
    printInfo_103012300239(Q);

    cout << "\nMelakukan pelayanan pada antrean:" << endl;
    serveQueue_103012300239(Q);

    cout << "\nIsi antrean setelah pelayanan:" << endl;
    printInfo_103012300239(Q);

    ElemQ* P6 = createElemQueue_103012300239("Edward", 22, "pekerja", 6, 9);
    enqueue_103012300239(Q, P6);

    cout << "\nMengatur ulang antrean berdasarkan prioritas:" << endl;
    reassignQueue_103012300239(Q);
    printInfo_103012300239(Q);

    cout << "\nMemeriksa waktu tunggu dan mengubah prioritas jika lebih dari 2 jam:" << endl;
    checkWaitingTIme_103012300239(Q, 130);

    printInfo_103012300239(Q);

    cout << "\nMenangani kondisi darurat untuk warga dengan nomor antrean 5:" << endl;
    emergencyHandle_103012300239(Q, 5);
    printInfo_103012300239(Q);

    cout << "\nMengupdate prioritas antrean setiap jam:" << endl;
    updatePriority_103012300239(Q);
    printInfo_103012300239(Q);

    cout << "\nMenghapus warga dengan nomor antrean 3:" << endl;
    ElemQ* removedElem = findAndRemove_103012300239(Q, 3);

    if (removedElem) {
```



```cpp
    if (removedElem) {
    cout << "Warga yang dihapus: " << info(removedElem).nama << endl;
    }

    printInfo_103012300239(Q);

    cout << "\nUkuran antrean saat ini: " << size_103012300239(Q) << endl;

    return 0;
}
```

Running



```
Isi antrean awal:
Daftar Antrean:
Nama: John Doe
Usia: 65
prioritas: Ya
Nomor Antrean: 1
----------------------------
Nama: Charlie
Usia: 70
prioritas: Ya
Nomor Antrean: 4
----------------------------
Nama: Bob
Usia: 25
prioritas: Tidak
Nomor Antrean: 3
----------------------------
Nama: David
Usia: 28
prioritas: Tidak
Nomor Antrean: 5
----------------------------

Melakukan pelayanan pada antrean:
Melayani warga:
Nama: John Doe
Usia: 65
Pekerjaan: lansia
Prioritas: Ya
Vaksinasi berhasil.
----------------------------
Melayani warga:
Nama: Charlie
Usia: 70
Pekerjaan: pensiunan
Prioritas: Ya
Vaksinasi berhasil.
----------------------------
Melayani warga:
Nama: Bob
Usia: 25
Pekerjaan: pekerja
Prioritas: Tidak
Vaksinasi berhasil.
----------------------------
Melayani warga:
Nama: David
Usia: 28
Pekerjaan: pekerja
Prioritas: Tidak
Vaksinasi berhasil.
----------------------------

Isi antrean setelah pelayanan:
Antrean KOSONG!!!

Mengatur ulang antrean berdasarkan prioritas:
Daftar Antrean:
Nama: Edward
Usia: 22
prioritas: Tidak
Nomor Antrean: 6
----------------------------

Memeriksa waktu tunggu dan mengubah prioritas jika lebih dari 2 jam:
Daftar Antrean:
Nama: Edward
Usia: 22
prioritas: Ya
Nomor Antrean: 6
----------------------------

Menangani kondisi darurat untuk warga dengan nomor antrean 5:
Warga dengan nomor antrean 5 tidak ditemukan.
Daftar Antrean:
Nama: Edward
Usia: 22
prioritas: Ya
Nomor Antrean: 6
----------------------------

Mengupdate prioritas antrean setiap jam:
Daftar Antrean:
Nama: Edward
Usia: 22
prioritas: Ya
Nomor Antrean: 6
----------------------------

Menghapus warga dengan nomor antrean 3:
Warga dengan nomor antrean 3 tidak ditemukan dalam antrean.Daftar Antrean:
Nama: Edward
Usia: 22
prioritas: Ya
Nomor Antrean: 6
----------------------------

Ukuran antrean saat ini: 1

Process returned 0 (0x0)   execution time : 0.095 s
Press any key to continue.
```