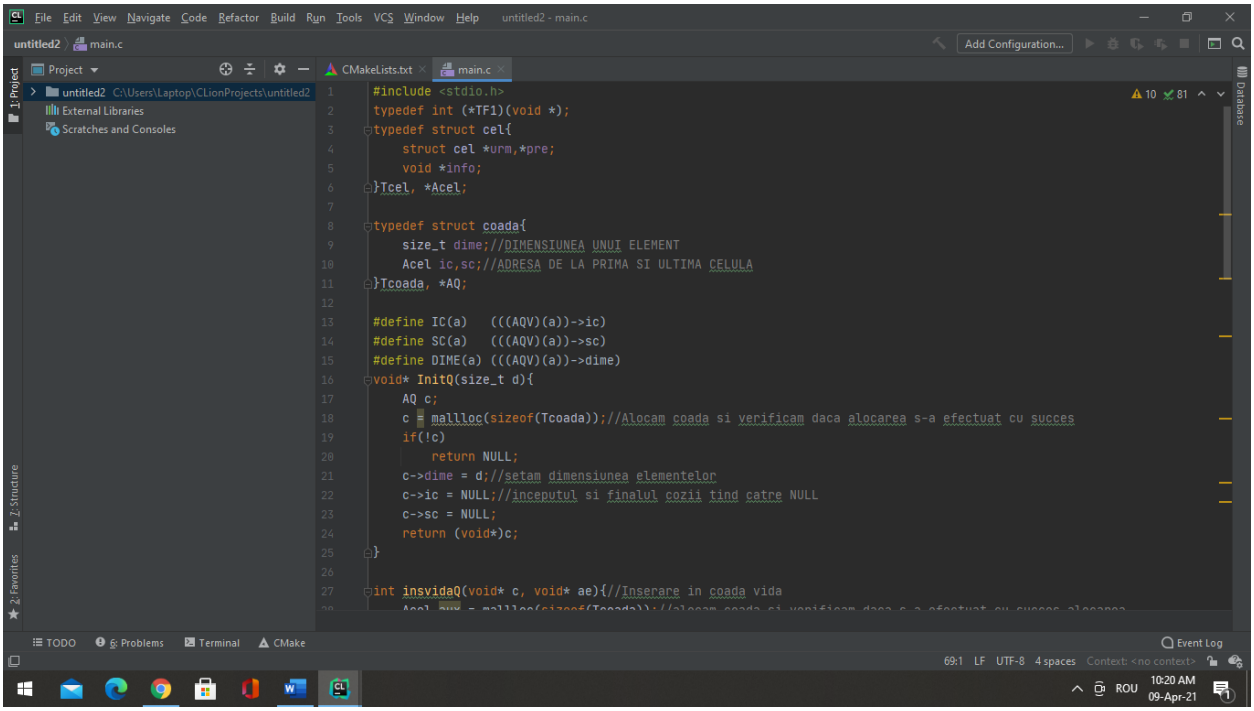
1)

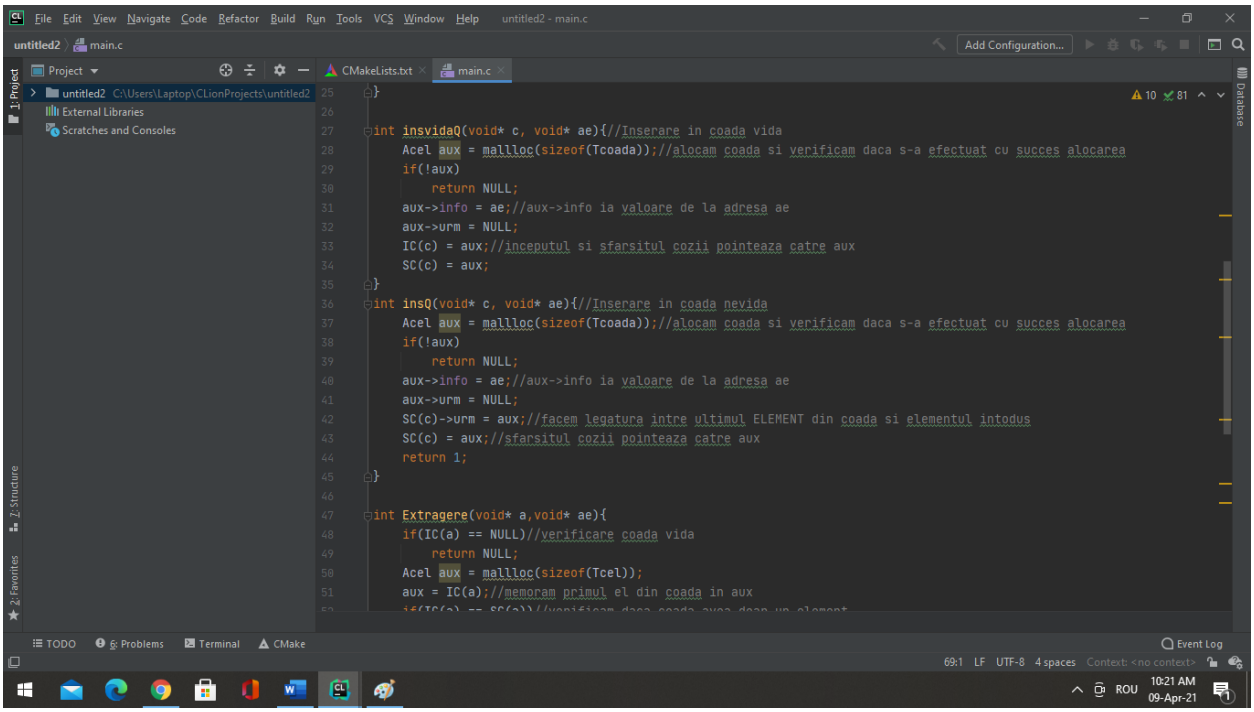


```c
#include <stdio.h>
typedef int (*TF1)(void *);
typedef struct cel{
    struct cel *urm,*pre;
    void *info;
}Tcel, *Acel;

typedef struct coada{
    size_t dime;//DIMENSIUNEA UNUI ELEMENT
    Acel ic,sc;//ADRESA DE LA PRIMA SI ULTIMA CELULA
}Tcoada, *AQ;

#define IC(a)   (((AQV)(a))->ic)
#define SC(a)   (((AQV)(a))->sc)
#define DIME(a) (((AQV)(a))->dime)
void* InitQ(size_t d){
    AQ c;
    c = malloc(sizeof(Tcoada));//Alocam coada si verificam daca alocarea s-a efectuat cu succes
    if(!c)
        return NULL;
    c->dime = d;//setam dimensiunea elementelor
    c->ic = NULL;//inceputul si finalul cozii tind catre NULL
    c->sc = NULL;
    return (void*)c;
}

int insvidaQ(void* c, void* ae){//Inserare in coada vida
```

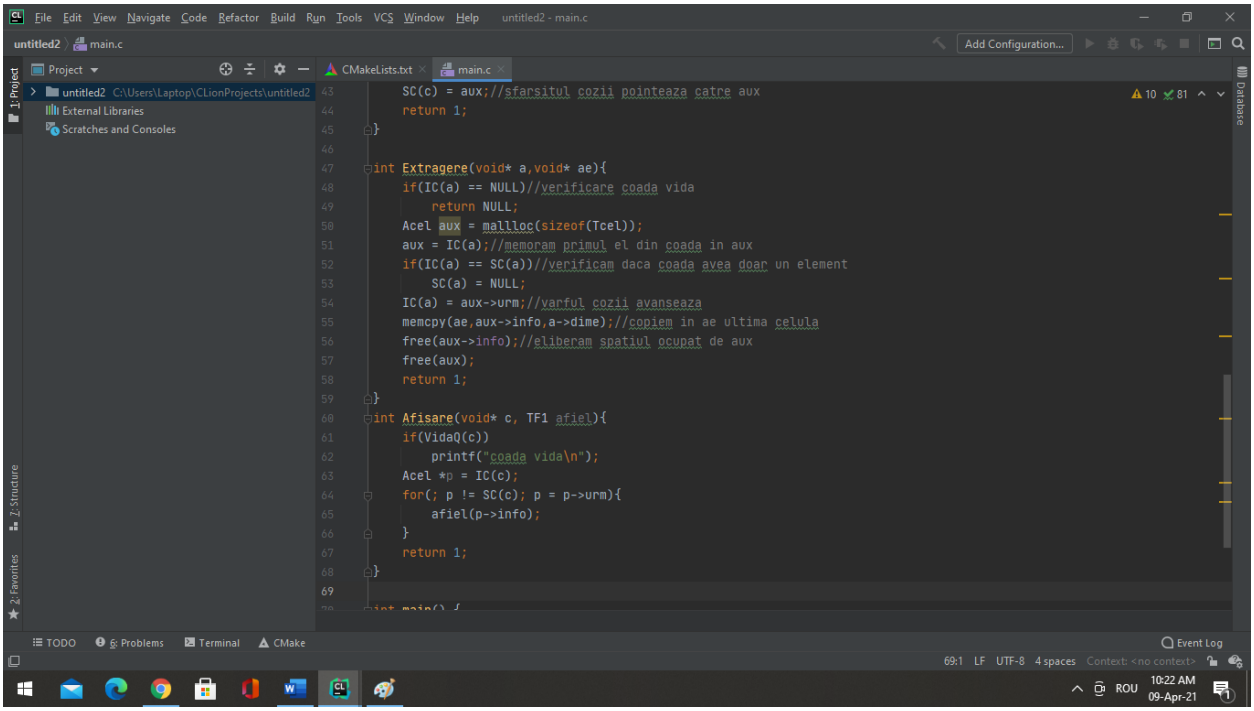


```c
}

int insvidaQ(void* c, void* ae){//Inserare in coada vida
    Acel aux = malloc(sizeof(Tcoada));//alocam coada si verificam daca s-a efectuat cu succes alocarea
    if(!aux)
        return NULL;
    aux->info = ae;//aux->info ia valoare de la adresa ae
    aux->urm = NULL;
    IC(c) = aux;//inceputul si sfarsitul cozii pointeaza catre aux
    SC(c) = aux;
}
int insQ(void* c, void* ae){//Inserare in coada nevida
    Acel aux = malloc(sizeof(Tcoada));//alocam coada si verificam daca s-a efectuat cu succes alocarea
    if(!aux)
        return NULL;
    aux->info = ae;//aux->info ia valoare de la adresa ae
    aux->urm = NULL;
    SC(c)->urm = aux;//facem legatura intre ultimul ELEMENT din coada si elementul intodus
    SC(c) = aux;//sfarsitul cozii pointeaza catre aux
    return 1;
}

int Extragere(void* a,void* ae){
    if(IC(a) == NULL)//verificare coada vida
        return NULL;
    Acel aux = malloc(sizeof(Tcel));
    aux = IC(a);//memoram primul el din coada in aux
```

1)


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help untitled2 - main.c
untitled2 main.c
Add Configuration...
Project
untitled2 C:\Users\Laptop\CLionProjects\untitled2
External Libraries
Scratches and Consoles
CMakeLists.txt
main.c
SC(c) = aux;//sfarsitul cozii pointeaza catre aux
return 1;
}
int Extragere(void* a,void* ae){
if(IC(a) == NULL)//verificare coada vida
return NULL;
Acel aux = mallloc(sizeof(Tcel));
aux = IC(a);//memoram primul el din coada in aux
if(IC(a) == SC(a))//verificam daca coada avea doar un element
SC(a) = NULL;
IC(a) = aux->urm;//varful cozii avanseaza
memcpy(ae,aux->info,a->dime);//copiem in ae ultima celula
free(aux->info);//eliberam spatiul ocupat de aux
free(aux);
return 1;
}
int Afisare(void* c, TF1 afiel){
if(VidaQ(c))
printf("coada vida\n");
Acel *p = IC(c);
for(; p != SC(c); p = p->urm){
afiel(p->info);
}
return 1;
}
TODO 6: Problems Terminal CMake
Event Log
69:1 LF UTF-8 4 spaces
ROU 10:22 AM 09-Apr-21