Zad1 (5). Zaimplementuj stos wykorzystując interfejs listy jednokierunkowej: PUSH, POP.
Zaimplementuj kolejkę wykorzystując interfejs listy jednokierunkowej: ENQUEUE, DEQUEUE.
Uzupełnij przygotowane fragmenty plików kolejkaxx.c oraz stosxx.c według wzorów z plików kolejka.txt stos.txt
Utwórz trzy pliki:
plik nagłówkowy z definicją strukturyi prototypami funkcji (patrz przykład)
Plik main.c na podstawie plików kolejka.txt, (stos.txt).
Oraz plik koleja.c (stos.c) zawierający interfejs kolejki (stosu)

Przykładowy fragment pliku stos.h:

```
#ifndef _STOS_H_
#define _STOS_H_

typedef struct tnode
{
        int value;
        struct tnode * next;

}node;


void insert_item_begin(node ** root, int val);
void show_stack(node * root);
void clear_stack(node ** root);
node * find_in_stack(node * root, int val);
//inne prototypy

#endif
```

Zad2 (5) Dane są następujące definicje :

```
 //Struktura z danymi studenta
typedef struct Osoba{
char* imie;
char* nazwisko;
}Osoba;
 //Struktura z listą dwukierunkową
typedef struct STUDENT Student
struct STUDENT{
Osoba dane;
Student* next;
Student* prev;};
```

Napisz program, który zapewni wykonanie podstawowych operacje na liście.

Aby wprowadzic dane napisz funkcję utwórz_element według schamatu :

```
Student* utworz_element()
{       Student *element;
        char bufor[30 + 1];
        char *imie, *nazwisko;
        printf("Podaj imie: ");
        gets(bufor);
        imie = …..... ????
        strcpy(imie, bufor);
           ….....................???????
        element = malloc(sizeof(Student));
        element->next = NULL;
        element->prev = NULL;
        element->dane.imie = imie;
        element->dane.nazwisko = nazwisko;

        return element;
}
void dodaj_na_poczatek(Student **head)
Student *temp =.......... }
```