#include <stdio.h>

#include <cs50.h>

#include <string.h>

#include <ctype.h>

typedef struct \_queue

{

char\* val;

struct \_queue\* prev;

struct \_queue\* next;

}

queue;

// глобальные переменные

queue\* head;

queue\* tail;

// прототипы функций

void enqueue (char\* word);

void create();

char\* dequeue();

void showqueue();

void destroy();

int main (void)

{

create();

enqueue("Anna");enqueue("Bob");enqueue("Gol");enqueue("NYT");enqueue("kloi");enqueue("HYret");enqueue("Gren2");showqueue();

char\* d = dequeue();char\* s = dequeue();printf("%s\n", d);printf("%s\n", s);showqueue();

enqueue("Min");showqueue();

destroy();showqueue();

}

void destroy()

{

queue\* ptr = head;

while (ptr != NULL)

{

ptr = ptr->next;

free(head);

head = ptr;

}

}

// напечатали весь список

void showqueue()

{

queue\* ptr = head;

while (ptr != NULL)

{

printf("%s ", ptr->val);

ptr = ptr->next;

}

printf("\n");

}

// убрали из очереди и вернули значение

char\* dequeue()

{

// проверяем, является ли элемент в очереди единственным

if (head->next == NULL)

{

char\* word = head->val;

free(head);

return word;

}

else

{

char\* word = head->val;

queue\* ptr = head->next;

free(head);

head = ptr;

head->prev = NULL;

return word;

}

}

// добавили в очередь

void enqueue (char\* word)

{

// проверяем, есть ли в очереди элементы

if (head->val == NULL)

{

head->val = word;

}

else

{

queue\* ptr = malloc(sizeof(queue));

ptr->val = word;

ptr->next = NULL;

ptr->prev = tail;

tail->next = ptr;

tail = ptr;

}

}

// создали очередь

void create()

{

queue\* ptr = malloc(sizeof(queue));

ptr->val = NULL;

ptr->next = NULL;

ptr->prev = NULL;

head = ptr;

tail = ptr;

}