Содержимое Makefile

```
all: prog


prog: prog.o part1.o part2.o

	gcc prog.o part1.o part2.o -o prog


prog.o: prog.c

	gcc -c prog.c


part1.o: part1.c

	gcc -c part1.c


part2.o: part2.c

	gcc -c part2.c


clean:

	rm -rf *.o prog
```

Содержимое part1.c

```
typedef struct node {
	int key;
	struct node *next;
}stack;

void push(stack **head, int key);
stack *pop(stack **head);
void print_stack(stack *head);
int size(stack *head);
void task(stack *head);
void menu();
```

Содержимое part2.c

```
#include "stdio.h"
#include "malloc.h"
#include "stdlib.h"

typedef struct node {
	int key;
	struct node *next;
}stack;

void push(stack **head, int key) {
	stack *tmp = malloc(sizeof(stack));
	if (tmp == NULL) {
```

```
                return;
        }
        tmp->next = *head;
        tmp->key = key;
        *head = tmp;
}

stack *pop(stack **head) {
        stack *out;
        if ((*head) == NULL) {
                return;
        }
        out = *head;
        *head = (*head)->next;
        return out;
}

void print_stack(stack *head) {
        if (head != NULL) {
                printf("%d  ", head->key);
                print_stack(head->next);
        }
}

int size(stack *head) {
        int k = 0;
        while (head) {
                k++;
                head = head->next;
        }
        return k;
}

void task(stack *head) {
        stack *tmp = NULL;
        int k = 0, i = 0, l, min;
        int a[100];
        k = size(head);
        while (head) {
                a[i] = head->key;
                free(head);
                head = head->next;
                i++;
        }
        for (i = 0; i < k; i++) {
                min = i;
                for (int j = i + 1; j < k; j++)
                        if (a[j] < a[min])
                                min = j;
                l = a[min];
                a[min] = a[i];
                a[i] = l;
        }
        for (int i = 0; i < k; i++) {
                push(&tmp, a[i]);
        }
}

void menu() {
        printf("========================\n");
        printf("||   1-Push           ||\n");
        printf("||   2-Pop            ||\n");
        printf("||   3-Print stack    ||\n");
```

```
	printf("||   4-Curry to task  ||\n");
	printf("||   5-Menu           ||\n");
	printf("||   0-End            ||\n");
	printf("==========================\n");
	printf("\n");
}
```

Содержимое prog.c

```
#include "stdio.h"
#include "malloc.h"
#include "stdlib.h"

typedef struct node {
	int key;
	struct node *next;
}stack;

int main() {
	stack *a = NULL;
	int n = 0, ch = 10, x = 0;
	menu();
	while (ch != 0) {
		printf("=> ");
		scanf("%d", &ch);
		switch (ch) {
		case 1:
			printf("Enter the element of the stack: ");
			scanf("%d", &n);
			push(&a, n);
			break;
		case 2:
			pop(&a);
			break;
		case 3:
			if (a) {
				print_stack(a);
				printf("\n");
			}
			else
				printf("Stack is empty!\n");
			break;
		case 4:
			task(a);
			break;
		case 5:
			menu();
			break;
		}
	}
	return 0;
}
```

```
Admin@LAPTOP-Q5U6S2UH:/mnt/c/Users/Admin/Desktop/Все для вуза$ make
gcc prog.o part1.o part2.o -o prog
Admin@LAPTOP-Q5U6S2UH:/mnt/c/Users/Admin/Desktop/Все для вуза$ ls
DjVuReader.lnk  part2.c  prog.o  Лабы  lab24.c  Makefile  part2.o  laba24.c
 part1.c  prog  laba26.c  part1.o  prog.c
Admin@LAPTOP-Q5U6S2UH:/mnt/c/Users/Admin/Desktop/Все для вуза$ ./prog
========================
||  1-Push                ||
||  2-Pop                 ||
||  3-Print stack         ||
||  4-Curry to task       ||
||  5-Menu                ||
||  0-End                 ||
========================
=> 1
Enter the element of the stack: 7
=> 1
Enter the element of the stack: 4
=> 1
Enter the element of the stack: 1
=> 1
Enter the element of the stack: 5
=> 1
Enter the element of the stack: 9
=> 3
9  5  1  4  7
=> 2
=> 3
5  1  4  7
=> 1
Enter the element of the stack: 3
=> 4
=> 3
7  5  4  3  1
=> 2
=> 3
5  4  3  1
```