Laboratorium 11

Cecylia Błaszczak

262129

Zadanie 1

List.c:

```
#include <stdarg.h>

#include <stdio.h>

#include <stdlib.h>


#include "list.h"


static Node_t * createNode(int head) {

        Node_t * node = (Node_t *) malloc(sizeof(node));

        node->head = head;

        node->tail = NULL;

        return node;

}void push(Node_t * root, int value) {

        Node_t * currentNode = root;

        if (currentNode != NULL) {

                while (currentNode->tail != NULL)

                        currentNode = currentNode->tail;

                currentNode->tail = createNode(value);}}

void printList(Node_t * root) {

        Node_t * currentNode = root;

        while (currentNode != NULL) {

                printf("%d\n", currentNode->head);

                currentNode = currentNode->tail;}}


Node_t * createList(unsigned int nodeCount, ...){

  va_list args;

  va_start(args , nodeCount);

  Node_t * root;

  for(int i = 0; i < nodeCount; i++){

      int value = va_arg(args , int);

      if(i == 0){

root = createNode(value);}

else {

      push(root,value);}  }

  va_end(args);

  return root;}
```

list.h:

```
#pragma once

#include "predicate.h"

typedef struct Node {
	int head;
	struct Node * tail;
} Node_t;

Node_t * createList(unsigned int nodeCount, ...);
void printList(Node_t * root);
void push(Node_t * root, int value);
void removeIf(Node_t ** root, Predicate predicate, int toCompare);
```

predicate.h:

```
#pragma once
#include <stdbool.h>
typedef bool (*Predicate)(int, int);
```

main.c:

```
#include <stdio.h>

#include "list.c"
int main() {
	// Create list
	const unsigned int nodeCount = 5;
	Node_t * root = createList(nodeCount, 2, 4, 6, 8, 10);
// Print created list
printf("Created new list:\n");
printList(root);
return 0;
```