**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Лабораторная работа №4

По дисциплине «Программирование»

 Списки.

Группа: АВТ-041

Студент: Дударь М.А.

Преподаватель: Балакин В.В.

НОВОСИБИРСК 2021

1. **Постановка задачи:**

Необходимо реализовать разность двух циклических списков. Операция разность – это результат, содержащий элементы из первой СД, которые отсутствуют во второй.

1. **Алгоритм программы:**
2. Заполнение односвязных списков случайными значениями.
3. Преобразование списков в циклические.
4. Реализация операцию разности на этих списках.
5. **Принцип работы:**

Структура узла списка.

typedef struct Node {  
 int value;  
 struct Node \*next;  
} Node;

Функция вставки нового элемента в список.

void push(Node \*\*head, int data) {  
 Node \*tmp = (Node \*) malloc(sizeof(Node));  
 tmp->value = data;  
 tmp->next = (\*head);  
 (\*head) = tmp;  
}

Функция зацикливания списка. Указатель на следующий узел последнего узла устанавливается на первый узел.

void loop(Node \*head) {  
 Node \*tmp = head;  
 if (tmp == NULL) {  
 return;  
 }  
 while (tmp->next) {  
 tmp = tmp->next;  
 }  
 tmp->next = head;  
}

Функция поиска элемента циклического списка предыдущего данному.

Node \*getPrev(Node \*head) {  
 Node \*tmp = head;  
 while (tmp->next != head) {  
 tmp = tmp->next;  
 }  
 return tmp;  
}

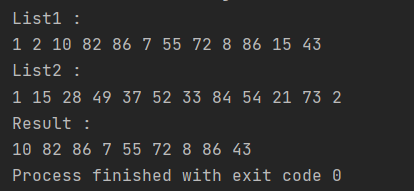
Функция разности двух циклических списков. Если находятся одинаковые элементы, то указатели списков перестраиваются и эти элементы освобождаются из памяти.

Node \*difference(Node \*head1, Node \*head2) {  
 Node \*tmp1 = head1;  
 Node \*tmp2 = NULL;  
 Node \*del = NULL;  
 int count = 0, flag;  
 do {  
 tmp2 = head2;  
 do {  
 del =NULL;  
 if (count == 1) {  
 count = 0;  
 head1 = tmp1;  
 }  
 if (tmp1->value == tmp2->value) {  
 Node \*buf = tmp1;  
 Node \*prev = getPrev(buf);  
 prev->next = tmp1->next;  
 if (tmp1 == head1) count = 1;  
 del = tmp1;  
 tmp1 = tmp1->next;  
 free(del);  
 flag = 256;  
 buf = tmp2;  
 prev = getPrev(buf);  
 del = tmp2;  
 prev->next = tmp2->next;  
 head2 = prev;  
 break;  
 } else flag = 0;  
 tmp2 = tmp2->next;  
 free(del);  
 } while (tmp2 != head2);  
 if (flag) continue;  
 tmp1 = tmp1->next;  
 } while (tmp1 != head1);  
 return head1;  
}

1. **Вывод:**

В ходе выполнения задания была произведена работа со списками на языке C. Организована система данных, реализовано взаимодействие с ними, добавление новых узлов, создание циклического списка, освобождение памяти.

Приложение 1. Вывод программы.



Приложение 2. Текст программы.

#include <stdio.h>  
#include <stdlib.h>  
#include <time.h>  
  
typedef struct Node {  
 int value;  
 struct Node \*next;  
} Node;  
  
void push(Node \*\*head, int data) {  
 Node \*tmp = (Node \*) malloc(sizeof(Node));  
 tmp->value = data;  
 tmp->next = (\*head);  
 (\*head) = tmp;  
}  
  
void loop(Node \*head) {  
 Node \*tmp = head;  
 if (tmp == NULL) {  
 return;  
 }  
 while (tmp->next) {  
 tmp = tmp->next;  
 }  
 tmp->next = head;  
}  
  
Node \*getPrev(Node \*head) {  
 Node \*tmp = head;  
 while (tmp->next != head) {  
 tmp = tmp->next;  
 }  
 return tmp;  
}  
  
Node \*difference(Node \*head1, Node \*head2) {  
 Node \*tmp1 = head1;  
 Node \*tmp2 = NULL;  
 Node \*del = NULL;  
 int count = 0, flag;  
 do {  
 tmp2 = head2;  
 do {  
 del =NULL;  
 if (count == 1) {  
 count = 0;  
 head1 = tmp1;  
 }  
 if (tmp1->value == tmp2->value) {  
 Node \*buf = tmp1;  
 Node \*prev = getPrev(buf);  
 prev->next = tmp1->next;  
 if (tmp1 == head1) count = 1;  
 del = tmp1;  
 tmp1 = tmp1->next;  
 free(del);  
 flag = 256;  
 buf = tmp2;  
 prev = getPrev(buf);  
 del = tmp2;  
 prev->next = tmp2->next;  
 head2 = prev;  
 break;  
 } else flag = 0;  
 tmp2 = tmp2->next;  
 free(del);  
 } while (tmp2 != head2);  
 if (flag) continue;  
 tmp1 = tmp1->next;  
 } while (tmp1 != head1);  
 return head1;  
}  
  
  
int main() {  
 Node \*head1 = NULL;  
 Node \*head2 = NULL;  
 Node \*tmp = NULL;  
 srand(time(NULL));  
 for (int i = 0; i < 10; ++i) push(&head1, rand() % 100);  
 loop(head1);  
 tmp = head1;  
 printf("List1 :\n");  
 do {  
 printf("%d ", tmp->value);  
 tmp = tmp->next;  
 } while (tmp != head1);  
 for (int i = 0; i < 10; ++i) push(&head2, rand() % 100);  
 loop(head2);  
 tmp = head2;  
 printf("\nList2 :\n");  
 do {  
 printf("%d ", tmp->value);  
 tmp = tmp->next;  
 } while (tmp != head2);  
  
 head1 = difference(head1, head2);  
 tmp = head1;  
 printf("\nResult :\n");  
 do {  
 printf("%d ", tmp->value);  
 tmp = tmp->next;  
 } while (tmp != head1);  
 return 0;  
}