НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Обчислювальної техніки та програмування»

**Звіт з практичної роботи №7**

Тема: «Подання рядків у пам'яті»

Виконав:

Єрмаков М.В.

КІТ-120в

Дата: 14.04.21

Харків 2021

**Варіант 5**

**Хід роботи**

**Завдання:**

Розробити функції, що забезпечують запис та читання запитів із пріоритетної черги, стека або дека.

В кожному завданні для організації вказаної черги використати дві структури. Перевірити працездатність розроблених функцій. Послідовність виконання операцій запису та читання обирати випадково.

Порівняти результати роботи, зробити висновки.

**Індивідуальне завдання:**

Стек. Стек організований на двоспрямованому списку та на масиві і «зростає» від меншої адреси пам’яті до більшої.

**Код:**

/\*\*

\* @author Ermakov M.

\* @date 26.04.2021

\* @brief EXERCISE №7

\*/

#include <iostream>

#include <stdlib.h>

using namespace std;

//структура списка с указателями на следующий и предыдущий элемент списка и с самим значением из структуры выше

typedef struct \_Node {

int k;

struct \_Node\* next;

struct \_Node\* prev;

} Node;

//хвост, голову и размер списка обьявляем в структуре DblLinkedList

typedef struct \_DblLinkedList {

size\_t size;

Node\* head;

Node\* tail;

} DblLinkedList;

void pushBack(DblLinkedList\* list, int value);

DblLinkedList\* createDblLinkedList();

void printDblLinkedList(DblLinkedList\* list);

int popBack(DblLinkedList\* list);

void stackOutput(int F);

int main() {

setlocale(LC\_ALL, "rus");

//регулируем длину списка

int F;

cout << "Stack length: " << endl;

cin >> F;

//вывод очереди

stackOutput(F);

return 0;

}

void stackOutput(int F) {

int\* arrayInt = new int[F];

for (int i = 0; i < F; ++i) //заполнение массива из F элементов числом 1

{

cin >> arrayInt[i];

}

//инициализируем список

DblLinkedList\* list = createDblLinkedList();

//заполняем стек элементами из массива

for (int i = 0; i < F; i++)

{

pushBack(list, arrayInt[i]);

}

//выводим список на экран

cout << "Stack output: ";

printDblLinkedList(list);

//выводим результат очистки стека

while(list != NULL)

cout << "Удален объект: " << popBack(list) << endl;

}

void printDblLinkedList(DblLinkedList\* list) {

Node\* tmp = list->head;

while (tmp) {

printf("%d ", tmp->k);

tmp = tmp->next;

}

printf("\n");

}

void pushBack(DblLinkedList\* list, int value) {

Node\* tmp = (Node\*)malloc(sizeof(Node));

if (tmp == NULL) {

exit(3);

}

tmp->k = value;

tmp->next = NULL;

tmp->prev = list->tail;

if (list->tail) {

list->tail->next = tmp;

}

list->tail = tmp;

if (list->head == NULL) {

list->head = tmp;

}

list->size++;

}

int popBack(DblLinkedList\* list) {

Node\* next;

int tmp;

if (list->tail == NULL) {

exit(4);

}

next = list->tail;

list->tail = list->tail->prev;

if (list->tail) {

list->tail->next = NULL;

}

if (next == list->head) {

list->head = NULL;

}

tmp = next->k;

free(next);

list->size--;

return tmp;

}

DblLinkedList\* createDblLinkedList() {

DblLinkedList\* tmp = (DblLinkedList\*)malloc(sizeof(DblLinkedList));

tmp->size = 0;

tmp->head = tmp->tail = NULL;

return tmp;

}

**Приклад результату виводу:**

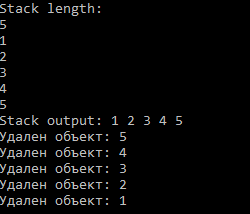
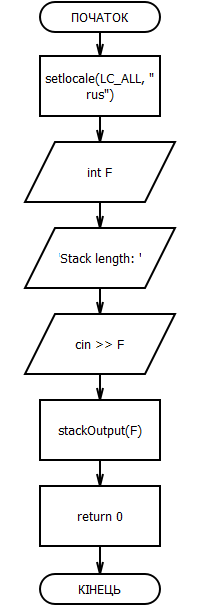


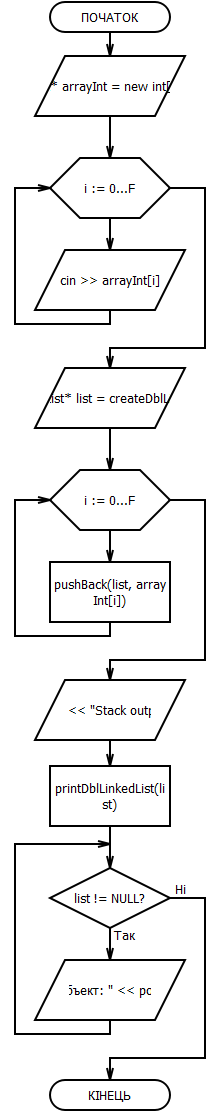
Рисунок 1 – Результат виводу програми

**Принцип дії програми:**

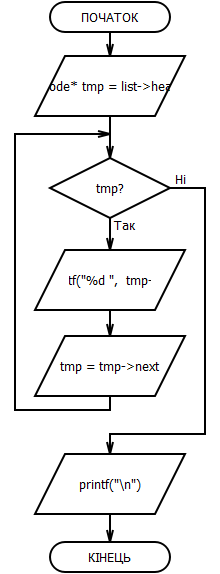
Перед початком програми створюємо структуру списку з наступним та попереднім показником й з цілим значенням. Починаємо виконання програми. Спочатку оголошуємо змінну типу int для визначення довжини нашого списку. Потім вводимо елементи масиву в список. Виводимо їх. Потім з кінця видаляємо кожний елемент. Виводимо кожне видалення в консоль.



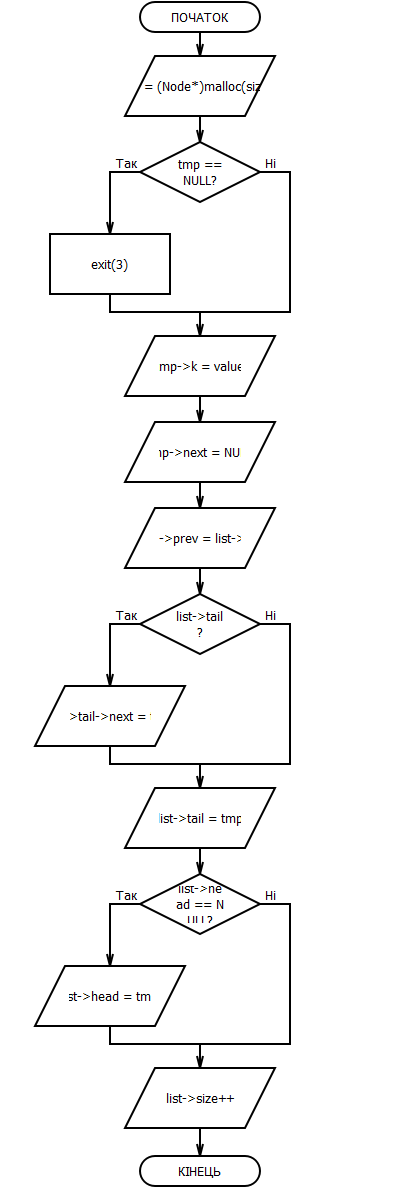
Блок-схема 1 – принцип роботи програми. Головна функція.



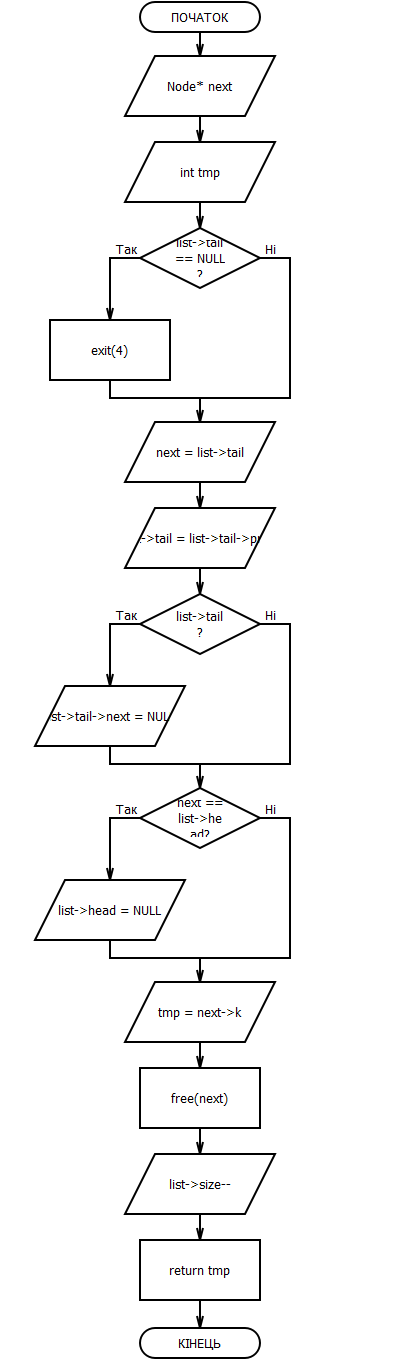
Блок-схема 2 – принцип роботи програми. Функція stackOutput()



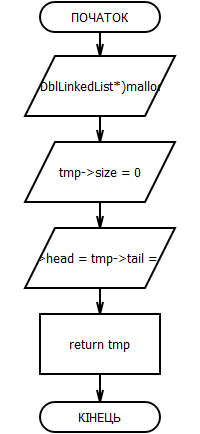
Блок-схема 3 – принцип роботи програми. Функція printDblLinkedList ().



Блок-схема 4 – принцип роботи програми. Функція pushBack().



Блок-схема 5 – принцип роботи програми. Функція popBack().



Блок-схема 6 – принцип роботи програми. Функція createDblLinkedList ().

**Висновок:** я набув практичного досвіду та закріпив знання про подання стека, дека, пріоритетної черги та дисципліни їх обслуговування.