 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №16**

з дисципліни «Алгоритмізація та програмування»

на тему: «Списки (стек, черга)»

Варіант №2

**Виконав:**

студент гр. БС-71

Батуркіна А.М.

**Перевірив:**

ас. каф. БМК

Рисін С. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2018

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

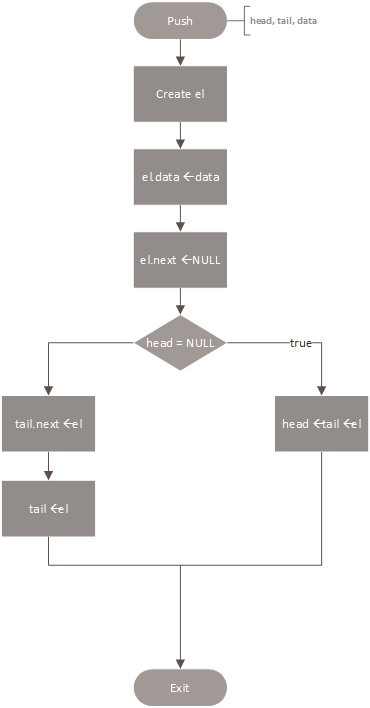
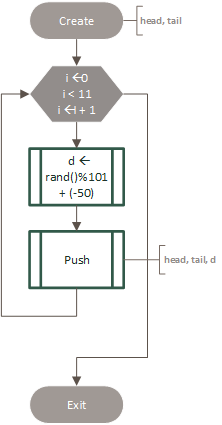
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

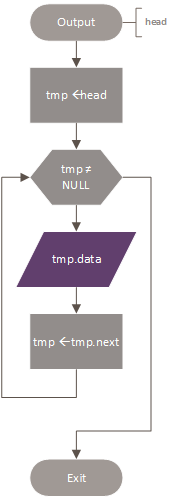
**Завдання:**

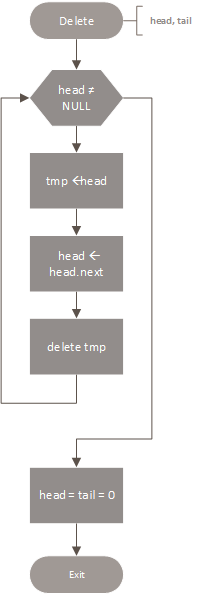
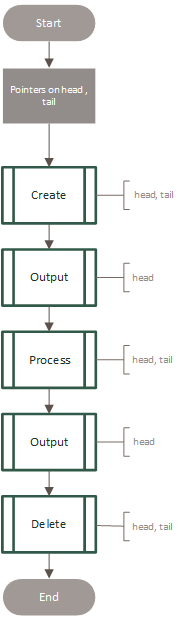
1. Ознайомитися з теоретичними відомостями створення, обробки та видалення однозв’язних списків та контейнерів, які створюються на їх основі.
2. Відповідно до варіанту визначити структуру стеку або черги та розробити функції, що необхідні для роботи із заданим контейнером: функцію додавання елемента push (пам’ять під черговий елемент контейнера слід виділяти динамічно), функцію вилучення елемента pop, функцію друку змісту контейнера на екран, функцію видалення контейнера.
3. Побудувати блок-схему алгоритму для вирішення задачі відповідно до свого варіанту.
4. Розробити програмний застосунок, в якому створюється контейнер і заповнюється елементами, що в якості даних зберігають унікальні числа діапазоном від **-50** до **50**, та реалізовано виконання індивідуального завдання, після чого всі створені контейнери видаляються (глобальні змінні не використовувати, функція main має бути призначена тільки для виклику функцій користувача):

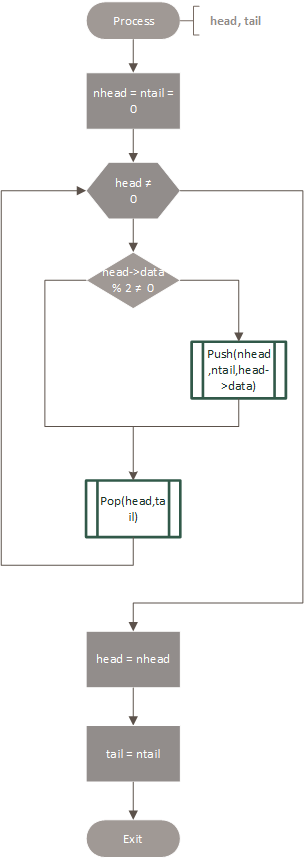
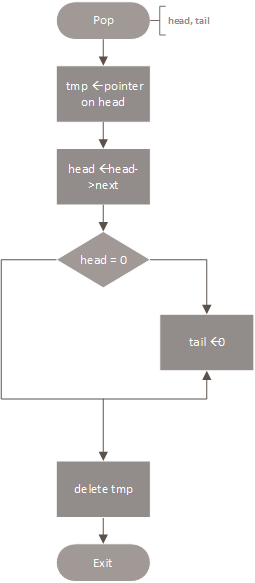
***Черга.*** *Видалити з черги всі парні числа.*

1. Скласти та захистити звіт по роботі.

**Блок-схема функції: **

****

**Блок-схема програми:**

****

**Лістинг**

#include<iostream>

#include<conio.h>

#include<ctime>

using namespace std;

struct queue

{

int data;

queue \*next;

};

void Push(queue \*&head, queue \*&tail, int data)

{

queue \*el = new queue;

el->data = data;

el->next = NULL;

if (head == NULL)

head = tail = el;

else

{

tail->next = el;

tail = el;

}

}

void Create(queue \*&head, queue \*&tail)

{

for (int i = 0; i< 11; i++)

{

int d = rand() % 101 + (-50);

Push(head, tail, d);

}

}

void Pop(queue \*&head, queue \*&tail)

{

queue \*tmp = head;

head = head->next;

if (head == NULL)

tail = NULL;

delete tmp;

}

void Process(queue \*&head, queue \*&tail)

{

queue \*ntail = 0, \*nhead = 0;

while (head)

{

if (head->data % 2 != 0)

Push(nhead, ntail, head->data);

Pop(head, tail);

}

head = nhead, tail = ntail;

}

void Output(queue \*head)

{

queue \*tmp = head;

while (tmp != NULL)

{

cout << tmp->data << " ";

tmp = tmp->next;

}

cout << endl;

}

void Delete(queue \*&head, queue \*&tail)

{

while (head != NULL)

{

queue \*tmp = head;

head = head->next;

delete tmp;

}

head = tail = 0;

}

int main()

{

srand(time(NULL));

queue \*head = 0, \*tail = 0;

Create(head, tail); // создаем очередь

Output(head); // вывод имэлементы

Process(head, tail); // выполняем задание

Output(head); // выводим элементы очереди после обработки

Delete(head, tail); // удаляем элементы новой очереди

\_getch();

}

**Результати:**

****

**Контрольні запитання:**

1. Як здійснюється зв’язок між елементами списку?

* Зв’язок між елементами здійснюються за допомогою посилання, які має кожен елемент(структура) списку.

1. Поясніть принцип роботи функції виведення на екран елементів списку?

* Створюється тимчасовий покажчик на голову для переміщення по списку. Далі в циклі(поки він не дорівнює нулю, тобто не є останнім елементом) відбувається виведення даних та присвоювання цьому покажчику значення покажчика на наступний елемент.

1. Чим відрізняються списки типу «стек» і «черга».

* Стек – такий контейнер, де для організації елементів використовується підхід: Last In First Out(LIFO), тобто видалення та додвання елементів можуть відбуватися тільки в/з голову/и списку. Черга – контейнер, спеціальний тип списку, в якому для організації елементів використовують підхід: First In First Out(FIFO), тобто видалення можливе лише з голови(початку черги), а додавання в кінець черги.

1. У чому переваги і недоліки стеку по відношенню до черги?

- Перевага полягає в швидкому доступі до останньо доданого елементу.

1. У чому переваги і недоліки двозв’язних списків по відношенню до однозв’язних?

* Перевага в тому, що по двозв’язному списку можна переміщатися в обидві сторони, через наявність посилання як на попередній, так і на наступний елемент. Недоліком можна вважати більш складну будову, що збільшує кількість необхідних махінацій при видаленні чи додаванні елементів.

1. Які існують обмеження на інформаційне поле для елементів списку, якщо в ньому зберігаються тільки value-значення?

- Обмежень нема( є лише в рамках типу даних).

1. З якою метою в програмах виконуються перевірка на порожність списку?

* Задля того, щоб не виконувати зайвих операцій, які не будуть мати сенс, якщо список порожній; для того, щоб зрозуміти, чи закінчився список.

1. З якою метою в програмах виконується видалення списків після завершення роботи з ними?

* Щоб звільнити пам’ять, так як після виконання програми нам більше не потрібна пам’ять, що була динамічно виділена для списку.

1. Які операції зі списками найчастіше доводиться виконувати на практиці?

* Додавання, вилучення, сортування, пошук, перестановка елементів, знаходження максимума/мінімума тощо.

1. Для моделювання яких реальних завдань зручно використовувати стек, а для яких чергу?

* Користь кожного типу списку(черга, стек) в унікальних властивостях. Для черги - FIFO, для стеку – LIFO.

Для стеку моделлю реального завдання може слугувати перехід по посиланням на сайті. Ми відкриваємо кожен раз новий сайт, створюючи список, а коли нам треба повернутися, ми закриваємо(видаляємо) останній елемент, який був доданий до цього списку.

Приклад черги – реальна черга людей чи якихось даних на обробку. Якщо людина перша стала в цю чергу, то вона повинна буде і піти перша. Так і з даними, першими обробляються ті, що перші увійшли.