 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 №17

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

доцент каф. БМК

к.т.н. Алхімова С.М.

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

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

(підпис викладача)

Київ-2019

**Завдання:**

1. Ознайомитися з теоретичними відомостями створення, обробки та видалення однозв’язних списків та контейнерів, які створюються на їх основі (стек, черга).
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):

***Стек : Визначити, скільки елементів стека знаходиться між його мінімальним і максимальним елементами.***

1. Скласти і захистити звіт по роботі.

**🞏 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:**

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

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

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

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

**Блок-схема:**









**Код:**

#include <iostream>

#include <conio.h>

using namespace std;

struct stack

{

int data;

struct stack\* next;

} \*top;

void push(int data\_to\_push)

{

struct stack\* temp;

temp = new stack();

if (!temp)

{

cout << "\nStack overflow!";

exit(0);

}

temp->data = data\_to\_push;

temp->next = top;

top = temp;

}

void pop()

{

struct stack\* temp;

if (top == NULL)

{

cout << "\nStack underflow!";

exit(0);

}

else

{

temp = top;

top = top->next;

temp->next = NULL;

free(temp);

}

}

void output()

{

struct stack\* temp;

if (top == NULL)

{

cout << "\nStack underflow!";

exit(0);

}

else

{

cout << "\nStack: ";

temp = top;

while (temp != NULL)

{

cout << temp->data << " ";

temp = temp->next;

}

cout << endl;

}

}

void delete\_stack()

{

struct stack\* temp;

temp = top;

while (top != NULL)

{

top = top->next;

temp->next = NULL;

}

free(temp);

cout << "Stack was fully deleted!";

}

void fill\_stack()

{

cout << "Numbers on the stack must be in the range [-50;50]\n";

bool x = 0;

do

{

cout << "Enter the number : ";

short num = 0;

cin >> num;

push(num);

cout << "\nDo You want to enter new number on the stack?\nNo - 0 or Yes - 1 : ";

cin >> x;

} while (x);

system("cls");

}

void min\_to\_max\_elements\_number()

{

fill\_stack();

output();

stack \*temp = top, \*min = top, \*max = top;

while (temp != NULL)

{

if (temp->data < min->data)

min = temp;

else

{

if (temp->data > max->data)

max = temp;

}

temp = temp->next;

}

temp = top;

int counter = 0;

bool flag = 0;

while (temp != NULL)

{

if (temp->next == min->next || temp->next == max->next)

{

flag = !flag;

temp = temp->next;

continue;

}

if (flag)

counter++;

temp = temp->next;

}

cout << "Minimum of the stack is " << min->data << '\n';

cout << "Maximum of the stack is " << max->data << '\n';

cout << "The number between Min and Max is " << counter << '\n';

delete\_stack();

output();

}

void main()

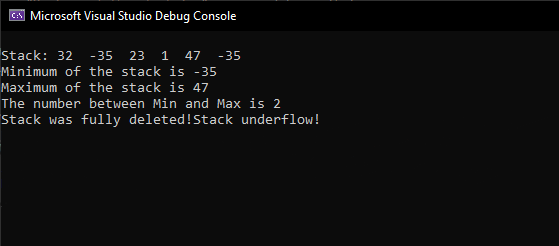
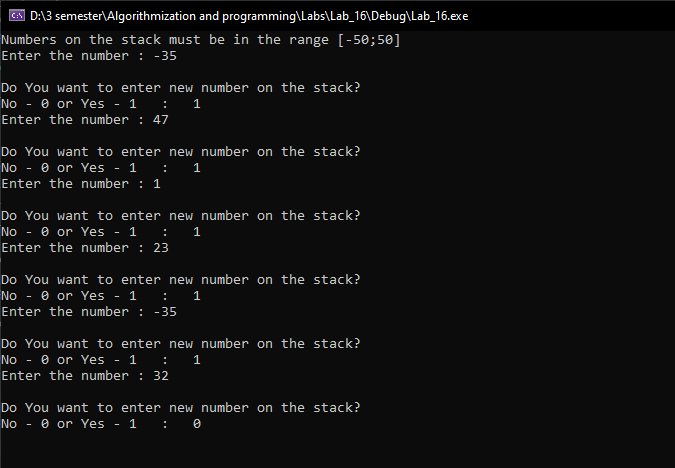
{

min\_to\_max\_elements\_number();

\_getch();

}

**Результати:**

****

**Контрольні питання:**

1. ***Як здійснюється зв'язок між елементами списку?***

Для зв’язку достатньо зберігати лише адресу першого елемнта. Кожному елементові треба “знати”, де розташовано елемент, який слідує за ним. Тоді до кожного елемента можна буде діставатися послідовно: від першого до другого, від другого до третього і т. д.

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

temp = top;

while (temp != NULL)

{

cout << temp->data << " ";

temp = temp->next;

}

У цьому прикаді, поки стек не закінчиться, виводиться зміст елементу списку, а сам вказівник на стек стає наступним за чергою вказівником цього стеку.

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

Черга відрізняється від стеку тим, що додавання здійснюється до одного кінця (хвоста), а вилучення - з протилежного (голови), у той час коли верхівка стеку і є останній доданий до стеку елемент.

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

Перевага – можливість «невидаляючого зчитування» останнього елементу.

Недолік – константний час доступу до елементів, O(N) у найгіршому випадку.

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

Перевага – доступ до даних з обох сторін списку, що дає свободу у переміщенні, додаванні, видаленні елементів у списках.

Недолік – необхідно зберігати більше покажчиків: на попередній і на наступний елементи.

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

Якщо немає ключа, то ми не можемо робити операцію порівняння за ключем. Обмеження можуть накладатися типом даних, що зберігаются у вузлах.

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

Якщо список порожній – неможливо робити деякі операції над ним, наприклад видалення елемента. З’явиться помилка пам’яті.

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

Видалення списків після завершення роботи з ними, робиться для того, щоб не перевантажувати пам'ять.

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

Функцію додавання елемента push, функцію вилучення елемента pop, функцію друку змісту контейнера на екран, функцію видалення контейнера.

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

Стек зручно використовувати для завдань, де потрібно внести або вилучити елементи з початку заданого завдання. Наприклад для завдання, де потрібно вирішити чи правильно розташовані в реченні дужки. При виявленні правої дужки з вершини стека витягується ліва дужка, вміщена останньою, і проводиться їх ідентифікація.

Чергу зручно використовувати для завдань де елементи вносяться з однієї сторони, а вилучаються з іншої. Її використовують для моделювання систем масової обслуговування: черги продуктів на полицях у супермаркетах, транспортних потоків, виробничих ліній і т.д.