**Шараев Владислав Николаевич 020602**

**Лабораторная работа №13**

**Вариант 14**

***Условие:*** Создать новый стек, в который поместить каждый третий элемент первого стека.

***Текст программы:***

#define \_CRT\_SECURE\_NO\_WARNINGS // Предпроцессор для отключения предупреждений при работе со scanf и printf

#include <iostream>

#include <cstdio>

using namespace std;

struct Stack { // Структура элемента стека

Stack\* parent = NULL; // Указатель на родительский элемент

int value; // Значение элемента

};

Stack\* AddStack(Stack\*, int); // Добавление элемента в стек

void FindThirdStack(Stack\*, int); // Поиск каждого третьего элемента и помещение в другой стек

void PrintStack(Stack\*); // Вывести все элементы стека на экран

void DeleteStack(Stack\*); // Удаление стека

Stack\* answer = new Stack; // Стек с каждым третьим элементом основного стека

int main()

{

setlocale(LC\_ALL, "Russian"); // Поддержка кириллицы

Stack\* base = new Stack; // Основной стек

int n;

printf("Введите количество элементов и элементы первого стека: ");

scanf("%d", &n);

for (int i = 0; i < n; i++) { // Чтение элементов основного стека

int temp;

scanf("%d", &temp);

base = AddStack(base, temp);

}

printf("Содержимое первого стека: \n");

PrintStack(base); // Вывод содержания основного стека на экран

FindThirdStack(base, n); // Поиск каждого третьего элемента

printf("Содержимое второго стека: \n");

PrintStack(answer); // Вывод содержания второго стека на экран

DeleteStack(base); // Удаление стека

return 0;

}

Stack\* AddStack(Stack\* item, int val) { // Добавление элемента в стек

Stack\* child = new Stack; // Создание дочернего элемента стека

child->parent = item; // Задаем родительский элемент стека

child->value = val; // Задаем значение элемента стека

return child; // Возвращаем указатель на новый последний элемент стека

}

void FindThirdStack(Stack\* item, int i) { // Поиск каждого третьего элемента и помещение в другой стек

if (item->parent != 0) {

FindThirdStack(item->parent, i - 1); // Рекурсивно идем по стеку, пока не дойдем до его начала

if (i % 3 == 0) { // Если номер элемента делится на 3, то добавляем его во второй стек

answer = AddStack(answer, item->value);

}

}

}

void PrintStack(Stack\* item) { // Вывести все элементы стека на экран

if (item->parent != NULL) {

printf("%d ", item->value);

PrintStack(item->parent); // Рекурсивно выводим все элементы стека

}

else {

printf("\n"); // Переход на новую строчку после вывода всех элементов

}

}

void DeleteStack(Stack\* base) { // Удаление стека

Stack\* item;

while (base != 0) {

item = base;

base = base->parent;

delete item;

}

}

***Результат работы программы:***



