МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Лабораторная работа № 2 по дисциплине «Технология программирования»

Выполнил студент группы ИВТб-21 _____/Монахов А.М/

Проверил преподаватель кафедры ЭВМ_____/Долженкова М.Л./

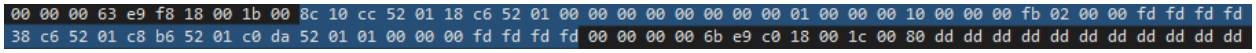
Киров 2021

1 Задание

Написать программу для работы с динамической структурой данных циклическая очередь, содержащей в каждом элементе массивы целых и вещественных чисел.

2 Работа программы

2.1 Занесение элемента



- Индикатор начала – $8C_{16}$ = 1000 1**1**00 (Выделенный бит – флаг занятости участка).
- Указатель на предыдущий занятый элемент.
- Указатель на следующий занятый элемент.
- Указатель на имя файла подкачки.
- Номер строки в файле подкачки.
- Тип участка памяти. (01 – пользовательский тип)
- Размер выделенного участка памяти.
- Количество обращений.
- Индикатор начала пользовательского участка памяти.
- Пользовательская структура из 16 байт.
  1. Указатель на массив типа int.
  2. Указатель на массив типа float.
  3. Указатель на следующий элемент.
  4. Размер массивов
- Индикатор конца пользовательского участка памяти.

2.2 Удаление элемента

Очистка массива:

```
fd fd fd f2 ea 41 18 00 10 00 80 dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd
dd dd dd dd dd dd dd dd f8 ea 4f 18 00 11 00 88 68 c4 52 01 28 c8 52 01 64 c3 dd 0f 5d 00 00 00 02 00 00 00 04 00 00 00 26 03 00
```

Очистка структуры

```
00 00 00 63 e9 f8 18 00 1b 00 80 dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd
dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd dd 00 00 00 00 6b e9 c0 18 00 1c 00 80 dd dd dd dd dd dd dd dd dd dd dd
```

Произошло освобождение области памяти после удаления элемента из дека, флаг занятости сменился на 0 - 80_{16} = 1000 0**0**00

3 Листинг программы

Листинг разработанной программы приведен в приложении А.

4 Вывод

В ходе выполнения лабораторной работы была разработана структура данных с использованием функций выделения (calloc) и освобождения (free) памяти – циклическая очередь. Освоена работа с дампом памяти в vs.

Приложение А
(обязательное)
Листинг программы

```
#include "pch.h"
#include <iostream>
#include <cstdlib>
#include <string>
using namespace std;

struct rec
{
	int* masi;
	float* masf;
	rec* next;
	int size;
};
template<class T>
T readnum(int min, int max, T type)
{
	bool fl = true;
	do {
		cin >> type;
		if (!cin.good() || cin.get() != '\n')
		{
			system("cls");
			cout << "Вы ввели неверное значение. Повторите ввод:\n ";
			cin.clear();
			cin.ignore(255, '\n');
		}
		else if (type < min || type > max) {
			system("cls");
			cout << "Вы ввели неверное число.\n";
		}
		else fl = false;
	} while (fl == true);
	return type;
}
int main()
{
	int num,nume,i;
	rec* P_first = NULL;
	rec* P;
	rec* P1;
	nume = 0;
	setlocale(LC_ALL, "Russian_Russia.1251");
	do {

		cout << ("\n");
		cout << ("1. Добавление элемента в очередь:\n");
		cout << ("2. Вывод всей очереди:\n");
		cout << ("3. Удаление первого элемента в очереди:\n");
		cout << ("4. Удаление всей очереди:\n");
		cout << ("5. Выход из программы:\n");
		cout << ("Выберите номер команды:\n");
		num = readnum(1, 5, 1);
		switch (num) {
		case 1: {
			if (nume == 0 )
		  {
			 P = (rec*)calloc(1,sizeof(rec));
		  std::cout << ("Введите количество элементов массива:\n");
```

```
                P->size = readnum(1, 100, 1);
                P->masi = (int*)calloc(P->size, sizeof(int));
                for (i = 0; i < P->size; i++)
                {
                        std::cout << "Введите " << i << " элемент массива: ";
                        P->masi[i] = readnum(-270000, 270000, 1);
                }
                P->masf = (float*)calloc(P->size, sizeof(float));
                for (i = 0; i < P->size; i++)
                {
                        std::cout << "Введите " << i << " элемент массива: ";
                        P->masf[i] = readnum(-10000000, 10000000, 1.0);
                }
                P->next = P;
                P_first = P;
                nume++;
                num = 0;
                break;
                     }
                     else {
                            P1 = (rec*)calloc(1, sizeof(rec));
                            cout << ("Введите количество элементов массива:\n");
                            P1->size = readnum(1, 100, 1);
                            P1->masi = (int*)calloc(P1->size, sizeof(int));
                            for (i = 0; i < P1->size; i++)
                            {
                                    cout << "Введите " << i << " элемент массива: ";
                                    P1->masi[i] = readnum(-270000, 270000, 1);

                            }
                            P1->masf = (float*)calloc(P1->size, sizeof(float));
                            for (i = 0; i < P1->size; i++)
                            {
                                    cout << "Введите " << i << " элемент массива: ";
                                    P1->masf[i] = readnum(-100000000, 100000000, 1.0);
                            }
                            P1->next = P_first;
                            P = P_first;
                            for (i = 1; i < nume; i++){
                                    P = P->next;
                            }
                            P->next = P1;
                            nume++;
                            num = 0;
                            break;
                     }

             }
             case 2: {
                    if (P_first != NULL) {
                            P = P_first;
                            for (int j = 0; j < nume; j++)
                            {
                                    cout << ("Массив целых чисел") << j + 1 << (" элемента
очереди:  ");
                                    for (i = 0; i < P->size; i++)
                                            cout << P->masi[i] << (", ");
                                    cout << ("\n");
                                    cout << ("Массив веественных чисел") << j + 1 << ("
элемента очереди:  ");
                                    for (i = 0; i < P->size; i++)
                                            cout << P->masf[i] << (", ");
                                    cout << ("\n");
                                    P = P->next;
```

```
                        }
                }
                else cout << ("Очередь пуста\n");

                num = 0;
                break;
        }
        case 3: { if (P_first != NULL) {
                P = P_first->next;
                free(P_first->masf);
                free(P_first->masi);
                free(P_first);
                P_first = P;
                nume--;
                cout << ("Элемент удален");
        } else cout << ("Очередь пуста\n");

                num = 0;
                break;
        }
        case 4: {  if (P_first != NULL) {
                for (i = 1; i < nume; i++)
                {
                        P = P_first->next;
                        free(P_first->masf);
                        free(P_first->masi);
                        free(P_first);
                        P_first = P;
                }
                free(P_first->masf);
                free(P_first->masi);
                free(P_first);
                P_first = NULL;
                nume = 0;
                cout << ("Очередь очищена");
        }  else cout << ("Очередь пуста\n");
                num = 0;
                break;
        }
        }
    } while (num != 5);
}
```