КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

ЛИНЕЙНЫЕ И ЦИКЛИЧЕСКИЕ СПИСКИ

по курсу: Структуры и алгоритмы обработки данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № _____ _____ _____
подпись, дата инициалы, фамилия

Санкт-Петербург 2022

ГУАП

Цель работы:

Целью работы является изучение структур данных «линейный

список» и «циклический список», а также получение практических навыков

их реализации.

4	Даны 2 многочлена. Каждый многочлен $P(x)=a_nx^n + a_{n-1}x^{n-1} +...+ a_1x + a_0$ с целыми коэффициентами можно представить в виде списка, причем если $a_i=0$, то соответствующее звено не включать в список. Определить процедуру, которая строит многочлен p – сумму многочленов q и r	Линейный односвязный

Задание на лабораторную работу:

Листинг программы:

main.cpp

```
#include <iostream>

using namespace std;

#include "libs/lib.h"

#include <cmath>
```

```
#include <time.h>



// проверка ввода

#include "libs/simple_char.h"

#include "libs/input_validation.h"



#include "structs.h"
```

```
list *get_func(int n, list *tmp, const char *promt = "") {

  int a;

  for (int i = n - 1; i >= 0; i--) {

    cout << promt << "[" << i << "] = ";

    a = read_value("", true, true, false);

    if (tmp == NULL) {
```

```
    tmp = create(i, a);



  } else {



    add_element_end(i, a, tmp);



  }



}









return tmp;
```

```
}



void draw_func(list *tmp, const char *promt = "") {


  cout << promt;


  bool first = true;


  while (tmp != NULL) {


    if (tmp -> a != 0) {
```

```
if (!first)

  cout << " + ";

else first = false;

cout << tmp -> a;
```

```
    if (tmp -> n != 0) cout << "x^" << tmp -> n;

  }

  tmp = tmp -> next;

}

cout << endl;
```

```
}



int get_size_list(list *tmp) {


  int size = 0;


  while (tmp != NULL) {


    size++;


    tmp = tmp -> next;
```

```
  }


  return size;


}


list *merge_func(list *a, list *b) {


  list *tmp = NULL;
```

```
int size_a = get_size_list(a);

int size_b = get_size_list(b);

if (size_b > size_a) {

  list *buf = a;

  a = b;

  b = buf;
```

```
}

while (a != NULL) {

  if (b != NULL && ((b -> n) == (a -> n))) {

    if (tmp == NULL) {

      tmp = create(a -> n, (a -> a) + (b -> a));
```

```
} else {

  add_element_end(a -> n, (a -> a) + (b -> a), tmp);

}

b = b -> next;

} else {
```

```
  if (tmp == NULL) {


    tmp = create(a -> n, a -> a);


  } else {


    add_element_end(a -> n, a -> a, tmp);


  }


}
```

```
    a = a -> next;



  }









  return tmp;



}









int main() {
```

```
    // смена кодировки


system("chcp 65001");


list *q_list = NULL;


list *r_list = NULL;


list *p_list = NULL;
```

```
  int n_q = read_value("Введите размер многочлена Q: ", true,
true, false);


  int n_r = read_value("Введите размер многочлена R: ", true,
true, false);


  q_list = get_func(n_q, q_list, "A");


  r_list = get_func(n_r, r_list, "B");


  draw_func(q_list, "Q(x) = ");
```

```
    draw_func(r_list, "R(x) = ");



    p_list = merge_func(q_list, r_list);

    draw_func(p_list, "P(x) = ");



        return 0;

}
```

stuct.h

```
#include <iostream>

using namespace std;

struct list {

  int n;
```

```
  int a;

  struct list *next = NULL;

};

list *create(int n, int a) {
```

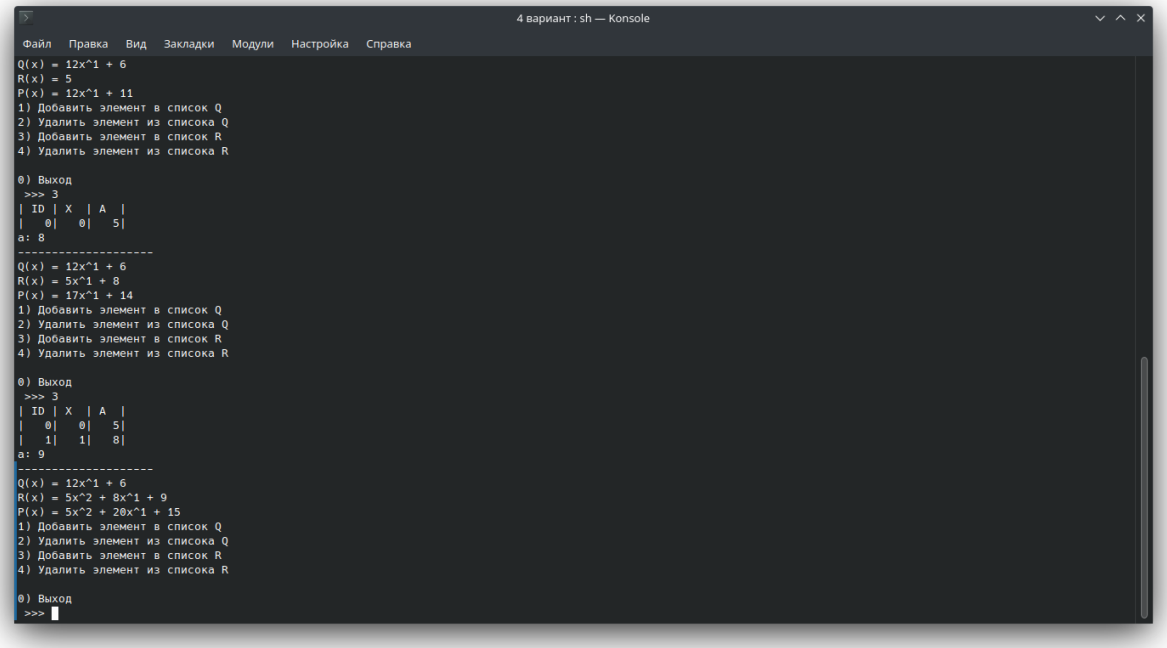
```
list *tmp = (list*)malloc(sizeof(list));

tmp -> n = n;

tmp -> a = a;

tmp -> next = NULL;

return tmp;
```

```
}



void add_element_end(int n, int a, list *head) {


  list *tmp = (list*)malloc(sizeof(list));


  tmp -> n = n;


  tmp -> a = a;


  list *p = head;
```

```
  while (p -> next != NULL)



  p = p -> next;



  p -> next = tmp;









}









void draw_list(list *number_list) {
```

```
  while (number_list != NULL){



    cout << number_list -> n << "|" << number_list -> a << endl;



    number_list = number_list -> next;



  }



}
```

Скриншот выполнения программы:


4 вариант : sh — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
Q(x) = 12x^1 + 6
R(x) = 5
P(x) = 12x^1 + 11
1) Добавить элемент в список Q
2) Удалить элемент из списока Q
3) Добавить элемент в список R
4) Удалить элемент из списока R
0) Выход
>>> 3
| ID | X | A |
| 0| 0| 5|
a: 8
Q(x) = 12x^1 + 6
R(x) = 5x^1 + 8
P(x) = 17x^1 + 14
1) Добавить элемент в список Q
2) Удалить элемент из списока Q
3) Добавить элемент в список R
4) Удалить элемент из списока R
0) Выход
>>> 3
ID	X	A
0	0	5
1	1	8
a: 9
Q(x) = 12x^1 + 6
R(x) = 5x^2 + 8x^1 + 9
P(x) = 5x^2 + 20x^1 + 15
1) Добавить элемент в список Q
2) Удалить элемент из списока Q
3) Добавить элемент в список R
4) Удалить элемент из списока R
0) Выход
>>>

Вывод: