Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы ИВТ-21-1б  
Мифтахов И.Г

Проверил  
ст. п. кафедры ИТАС  
Яруллин Д.В.

Пермь 2022

Постановка задачи:

1. Сформировать однонаправленный и двунаправленный списки или стек и очередь. Тип информационного поля указан в варианте.

2. Распечатать полученную структуру.

3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

5. Удалить соответствующую структуру из памяти.

Анализ задачи:

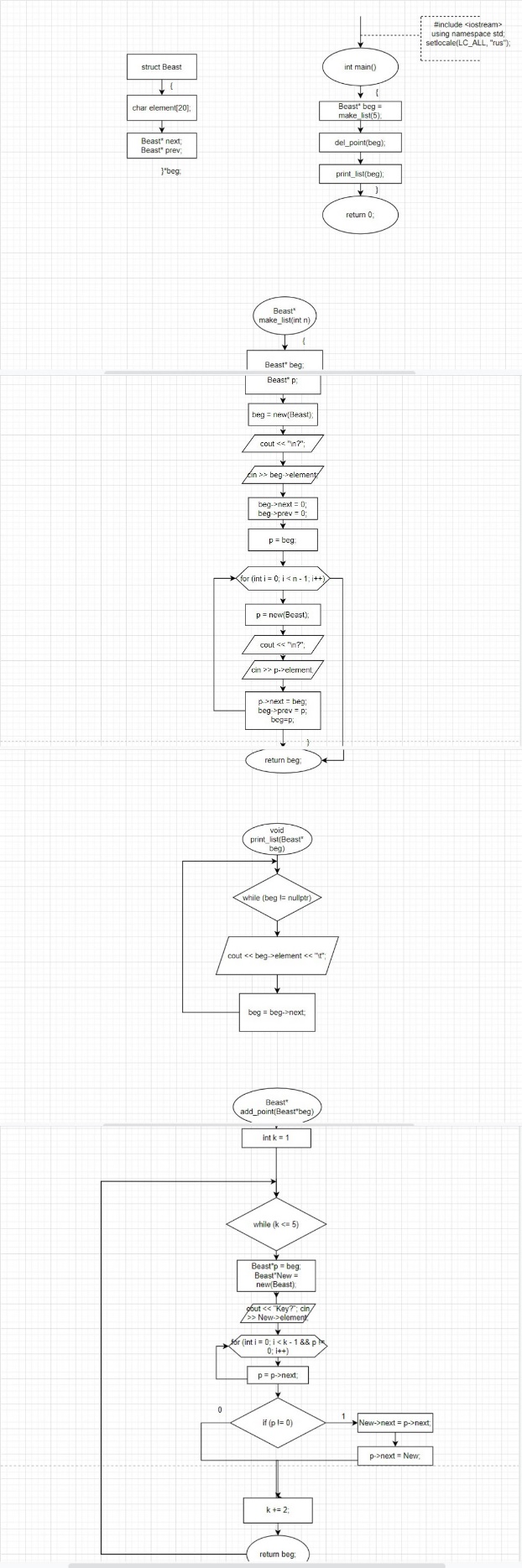
Создаем структуру Beast, в которой будет инфополе типа char и указатель типа структуры на следующий элемент и на предыдущий.

Далее создаем сам список через отдельную функцию, передав ей заранее количество элементов списка. Создаем указатели на начало списка и два вспомогательных. Создаем первый узел, заносим значение, очищаем его адресное поле, куда потом будет записан адрес следующего узла. Ставим указатель p на последний(первый) элемент и входим в цикл for. Создаем новый элемент, очищаем его адресное поле, а в адресное поле первого узла вносим адрес этого узла. Список создан.

Добавлять элементы следует в отдельной функции. Обозначаем номер первого добавленного элемента. Входим в цикл while, аналогичный первому. Ставим указатель на начало списка, создаем новый элемент. доходим до предшествующего элемента нужному элементу через цикл for. И если нужный элемент существует, тогда - Ставим на него указатель, доходим до следующего элемента после нужного и ставим другой указатель. Связываем их и искомый элемент добавлен. В конце увеличиваем k на 2.

Печатать будем в отдельной функции. В цикле while, пока у нас адрес следующего элемента не будет равен нулю будем выводить элементы узлов, переходя к следующим.

Блок-схема:



Код программы:

#include<iostream>

using namespace std;

struct Beast

{

char element[20];

Beast\* next;

Beast\* prev;

}\*beg;

Beast\* make\_list(int n)

{

Beast\* beg;//указатель на первый элемент

Beast\* p;//вспомогательные указатели

beg = new(Beast);//выделяем память под первый элемент

cout << "\n?";

cin >> beg->element;//вводим значение информационного поля

beg->prev = 0;

beg->next = 0;//обнуляем адресное поле

//ставим на этот элемент указатель p (последний элемент)

p = beg;

for (int i = 0; i < n - 1; i++)

{

p = new(Beast);//создаем новый элемент

cout << "\n?";

cin >> p->element;

p->next = beg;

beg->prev = p;//связываем p и r

//ставим на r указатель p (последний элемент)

beg=p;

}

return beg;//возвращаем beg как результат функции

}

void print\_list(Beast\* beg)

//печать списка

{

while (beg != nullptr)

{

cout << beg->element << "\t";

beg = beg->next;//переход к следующему элементу

}

}

Beast\* add\_point(Beast\*beg)

{

int k = 1;

while (k <= 5)

{

Beast\*p = beg;//встали на первый элемент

Beast\*New = new(Beast);//создали новый элемент

cout << "Key?"; cin >> New->element;

for (int i = 0; i < k - 1 && p != 0; i++)

p = p->next;//проходим по списку до(k-1) элемента или до конца

if (p != 0)//если k-й элемент существует

{

New->next = p->next;//связываем New и k-й элемент

p->next = New;//связываем (k-1)элемент и New

}

k += 2;

}

return beg;

}

int main()

{

setlocale(LC\_ALL, "Russian");

Beast\* beg = make\_list(5);

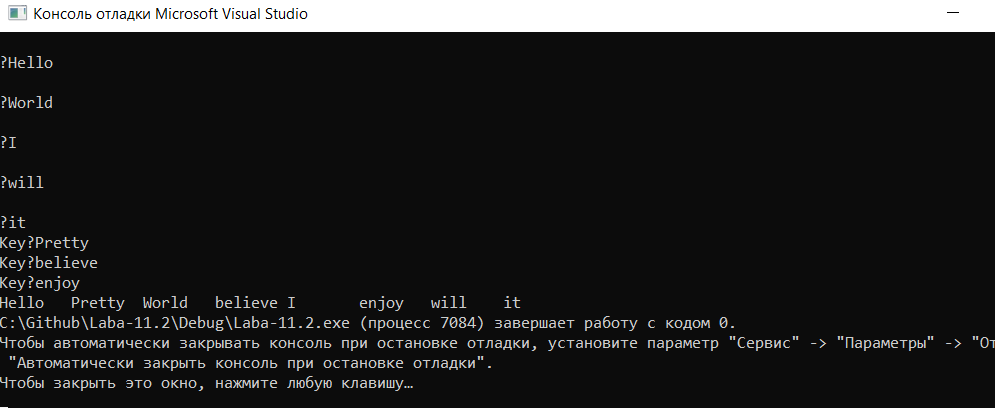
add\_point(beg);

print\_list(beg);

return 0;

}

Скриншоты результатов:



Анализ результатов:

Программа работает корректно. Выводится список с добавленными элементами.