**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Кафедра безопасности информационных систем**

**ОТЧЁТ**

по практической работе работе №7 на тему:   
**«Разработка и применение циклического списка.**

**Задача Иосифа.»**

по дисциплине «Алгоритмы и структуры данных»

Выполнил: студент группы ИСТ-114,Константинов А.Д.

«\_\_» \_\_\_\_\_\_\_\_ 2022 г. \_\_\_\_\_\_\_\_\_\_\_/А.Д. Константинов/

Принял: к.ф.-м.н., доцент, И.А. Моисеев

« » ноября 2022 г. \_\_\_\_\_\_\_\_\_\_\_/ И.А. Моисеев /

**Содержание:**

[Цель работы: 3](#_Toc114692552)

[Листинг кода: 3](#_Toc114692553)

[Результат работы программы: 8](#_Toc114692554)

[Вывод: 8](#_Toc114692555)

[Приложение: 22](#_Toc114692556)

Цель работы:

# Пусть общее число участников 20n = , а расчет m производится от 2 и до 20. На экран монитора требуется

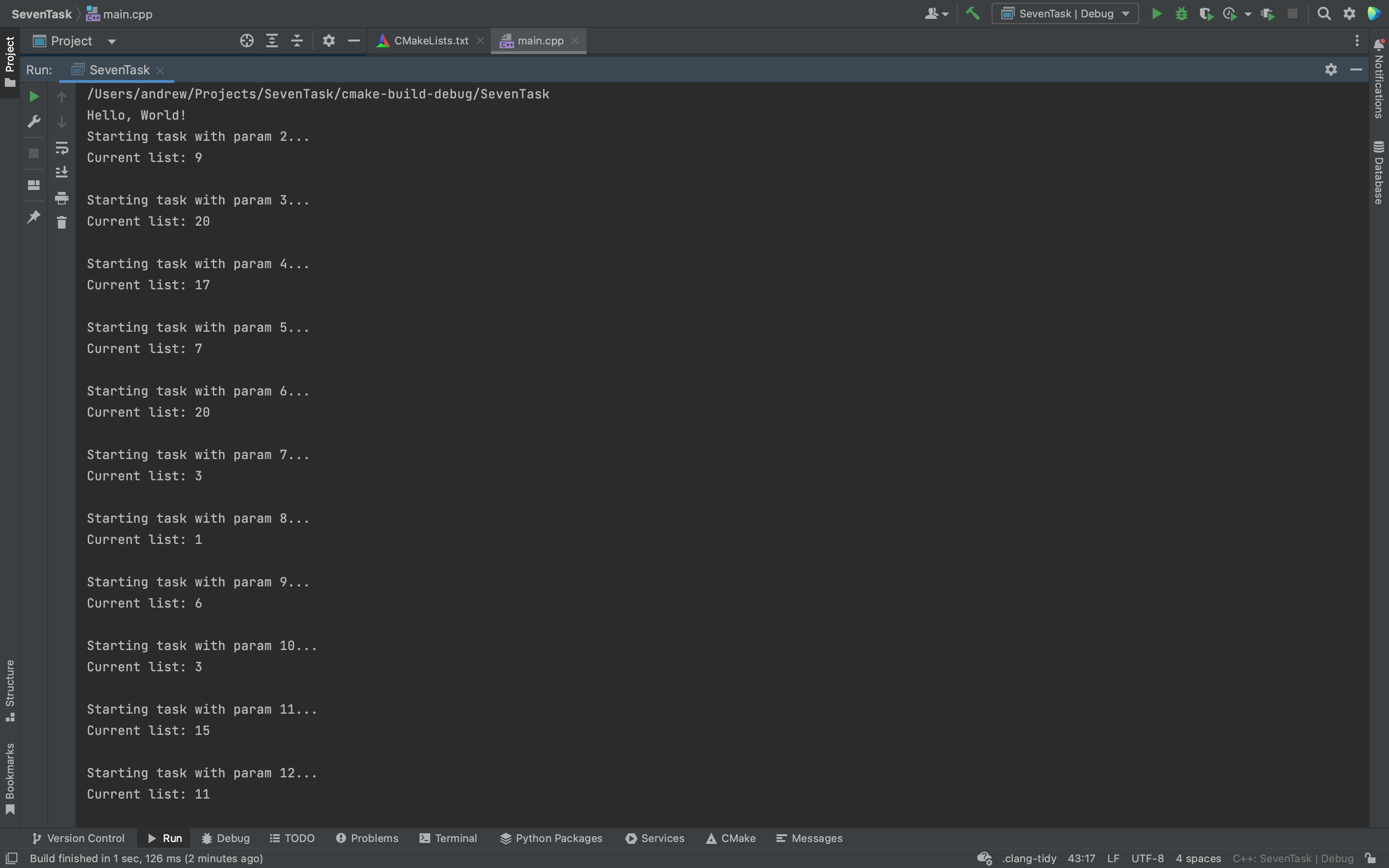
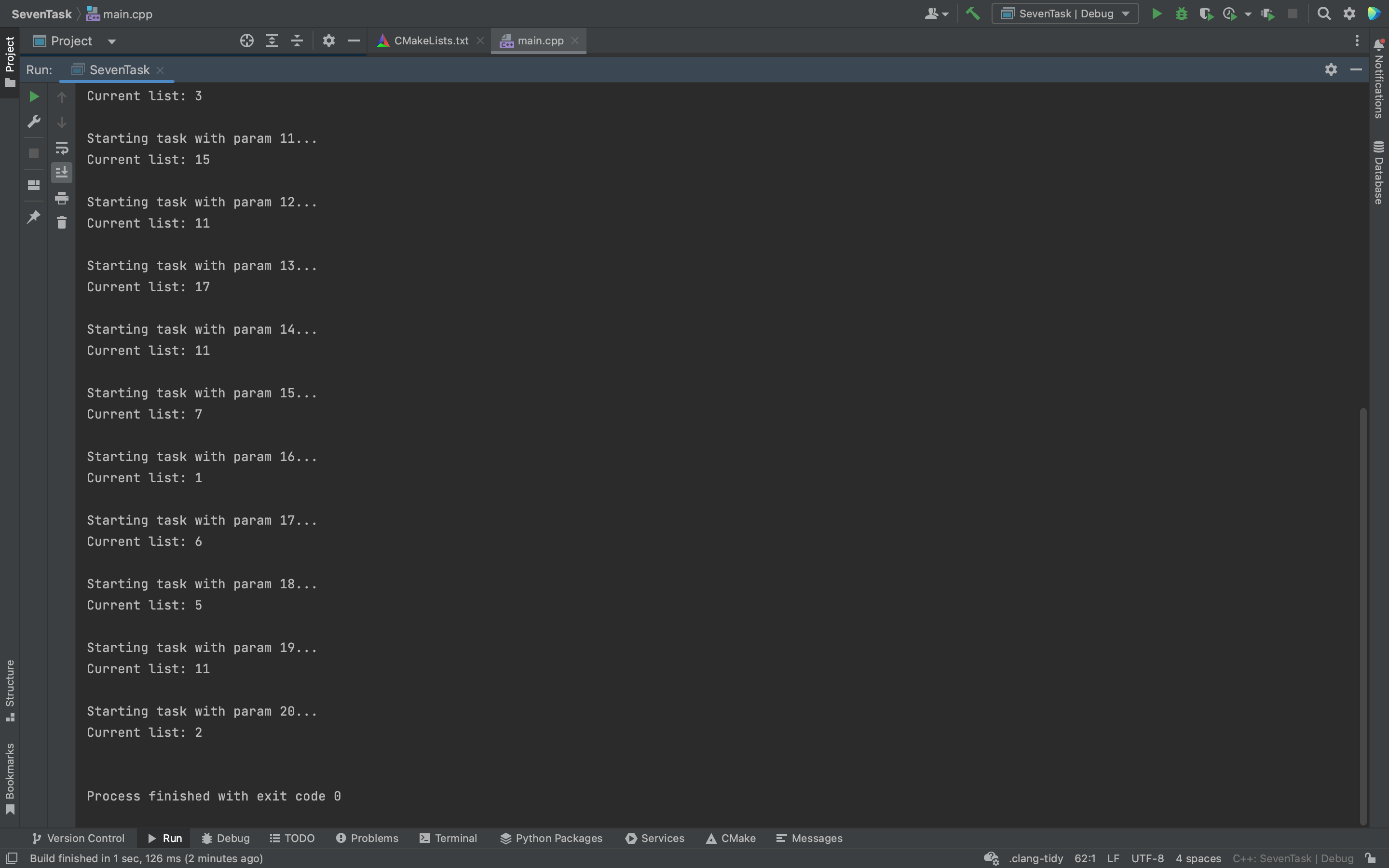
# вывести 19 результатов расчета, т.е. n фиксированное число и при каждом расчете равно 20, а m меняется

# от 2 при первом до 20 при последнем расчете.

# Листинг кода:

#include <iostream>  
  
struct ListNode{  
 ListNode\* next;  
 int value = 0;  
};  
void ListNodeCreate(ListNode\*\* listNode){  
 (\*listNode)->next = \*listNode;  
}  
void ListNodePush(ListNode\*\* listNode,int value){  
  
 ListNode\* newNode = new ListNode;  
 newNode->value = value;  
 auto v = (\*listNode)->value;  
 if((\*listNode)->next == (\*listNode) && (\*listNode)->value == 0){ // Если лист пустой  
 \*listNode = newNode;  
 (\*listNode)->next = newNode;  
 }  
 else{  
 auto test = \*listNode;  
 while(test->next != \*listNode) test = test->next;//Прокручиваем список до конца  
  
 auto prevAddr = test;  
 auto nextAddr = \*listNode;  
 newNode->next = nextAddr;  
 prevAddr->next = newNode;  
 }  
}  
void ListNodeDeleteByCount(ListNode\*\* listNode, int counts){  
 auto currentAddr = \*listNode;  
 auto prevAddr = \*listNode;  
 for (int i = 1; i < counts; ++i) {  
 prevAddr = currentAddr;  
 currentAddr = currentAddr->next;  
 }  
 prevAddr->next = currentAddr->next;  
 \*listNode = currentAddr->next;  
 delete(currentAddr);  
}  
void ListNodePrint(ListNode \* listNode){  
  
 std::cout << "Current list: " << (listNode)->value <<" ";  
 auto curr = (listNode)->next;  
 while(curr != listNode){  
 std::cout << curr->value << " ";  
 curr = curr->next;  
 }  
 std::cout << std::endl;  
}  
int ListNodeLen(ListNode \* listNode){  
 if(listNode == listNode->next) return 0;  
 int result = 1;  
 auto curr = listNode->next;  
 while(curr != listNode){  
 result++;  
 curr = curr->next;  
 }  
 return result;  
}  
int main() {  
 std::cout << "Hello, World!" << std::endl;  
 for (int i = 2; i <= 20; ++i) {  
 ListNode\* listNode = new ListNode;  
 ListNodeCreate(&listNode);  
 for (int j = 1; j <= 20; ++j) {  
 ListNodePush(&listNode,j);  
 }  
 std::cout << "Starting task with param " << i << "..." << std::endl;  
 while(ListNodeLen(listNode) >= 1){  
 ListNodeDeleteByCount(&listNode,i);  
 }  
 ListNodePrint(listNode);  
 std::cout<< std::endl;  
 delete(listNode);  
 }  
 return 0;  
}

# Результат работы программы:

# Вывод:

В ходе выполнения практической работы была написана программа, работающая с циклическим списком для решения задачи Иосифа