Министерство высшего образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

по лабораторной работе №7

Тема: «Перегрузка функций и функции с переменным числом параметров»

Выполнил

Студент группы РИС-22-2б

Будин Д.В.

Проверил доц. Кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2022

**7.1. Перегрузка функций.**

Написать перегруженные функции и основную программу, что их выполняет.

24. а) для массива целых чисел удаляет все четные элементы из массива;

б) для строки удаляет все четные слова.

**Анализ**

Для начала у пользователя запрашивают тип данных на ввод: строка/число, ответ заносится в булевую перенную flag. Если строка – 1, иначе 0.

* Если строка, то она передается в функцию типа char\* toNewArrive.

char\* result = toNewArrive(row);

1. Данная строка передается в функцию createListWords из предыдущей лабораторной работы, которая создает массив слов, а также считает кол-во слов(передается через ссылку).
2. Чтобы вернуть из функции массив, создается указатель на динамический массив char\*.
3. Через два вложенных цикла заполняется строка:

внешний – по кол-ву слов с шагом 2: for (int i = 0; i < count\_words; i += 2)

внутренний – по буквам в слове: for (int j = 0; words[i][j] != '\0'; j += 1)

result[count\_letter] = words[i][j], где count\_letter увеличивается на 1 на каждой итерации внутреннего и внешнего цикла(т.к. после слова ставится пробел result[count\_letter] = ' ';).

1. result[count\_letter] = '\0';
2. Очищается память под массив слов через ф-ию deleteListWords и возвращается результат.
3. cout << result << endl – вывод готовой строки
4. delete[] result – очистка памяти;

* Если число, то у пользователя запрашивают, что нужно убрать: четные числа или четные индексы. Считывается кол-во чисел в массиве, сам массив.

Функции реализуют разный код, но их тип один и тот же \*int.

Если индексы: new\_arr = toNewArrive(arr, n)

1. int\* result = new int[(n + 1) / 2] – создание динамического массива под числа, (n+1) / 2 – размерность с учетом четности кол-ва чисел.

5: [1, 2, 3, 4, 5] -> [1, 3, 5] – 3, 6: [1, 2, 3, 4, 5, 6] -> [1, 3, 5] -> 3;

1. Т.к. индексы у основного массива и созданного разные, то заноситься данные будут по счетчику j; j = 0;
2. Перебор чисел из основного массива с шагом 2: for (int i = 0; i < n; i += 2)

и занесение данных: result[j] = arr[i]; j += 1;

1. Изменение кол-во чисел в массиве: n = (n + 1) / 2;
2. Возврат массива, вывод массива с 0 по n индекс, измененный выше.

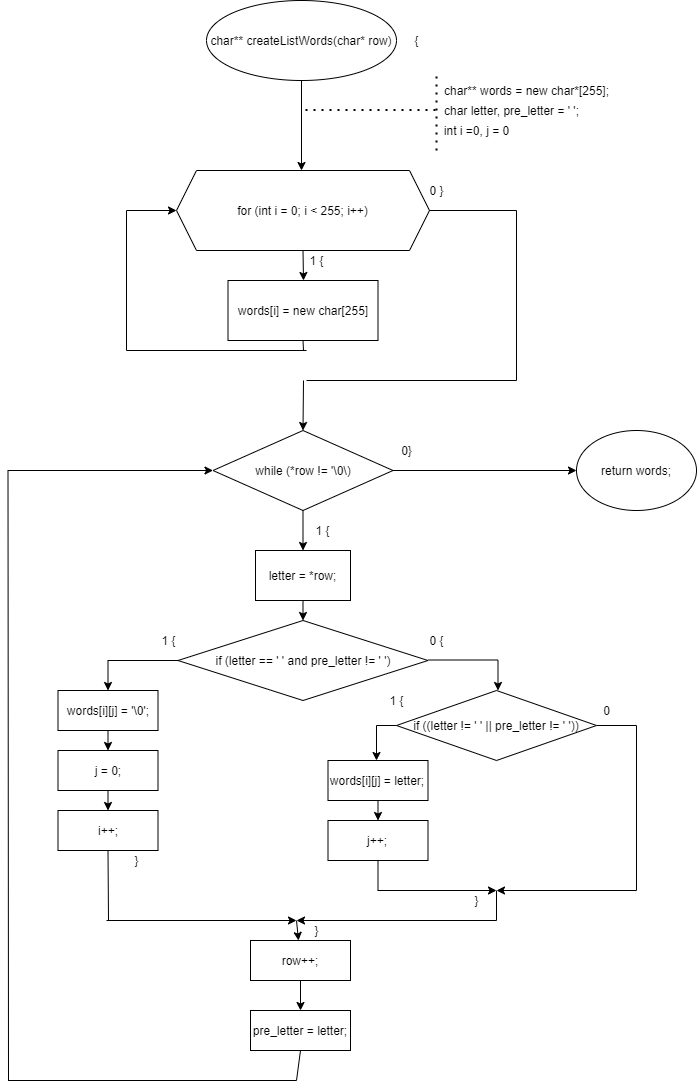
Если числа: new\_arr = toNewArrive(n, arr)

1. Int x = 0 – счетчик удаленных четных чисел;

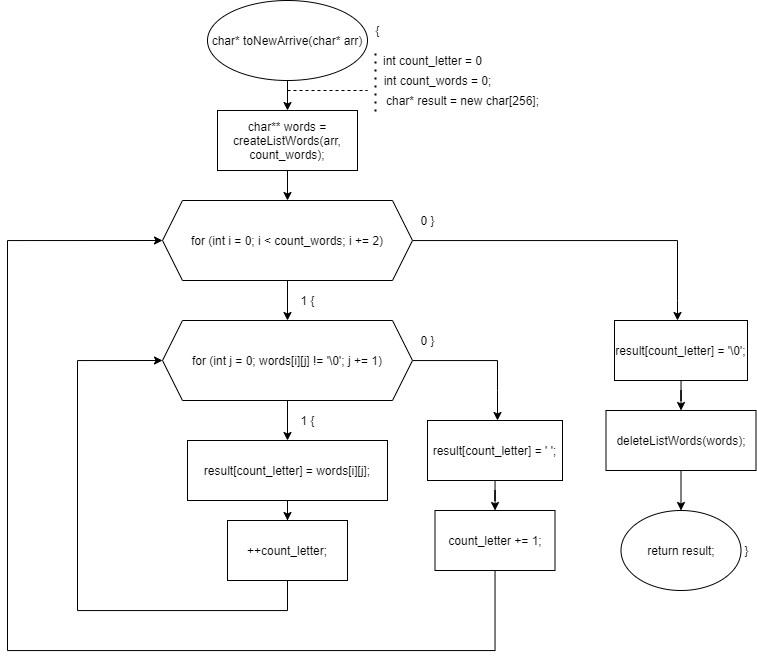
Через два вложенных цикла осуществляется перенос нечетных чисел в начало массива: если число четное - ++x, сдвиг через цикл for (int j = i + 1; j < n; j++) arr[j - 1] = arr[j];

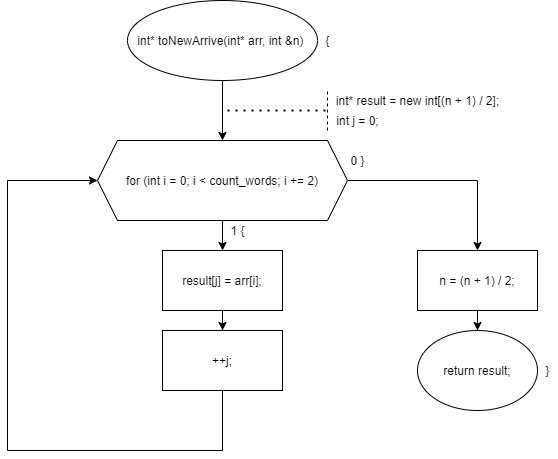
1. Тк удалили число: --i;
2. n -= x – уменьшение размерности итогового массива на кол-во четных чисел.
3. Массив готов к выводу, но в конце массива мусор
4. Заносим четные числа в новый динамический массив с новой размерностью int\* result = new int[n];
5. for (int i = 0; i < n; i++) result[i] = arr[i];
6. Возврат массива, вывод массива с 0 по n индекс, измененный выше.

**Алгоритм программы**

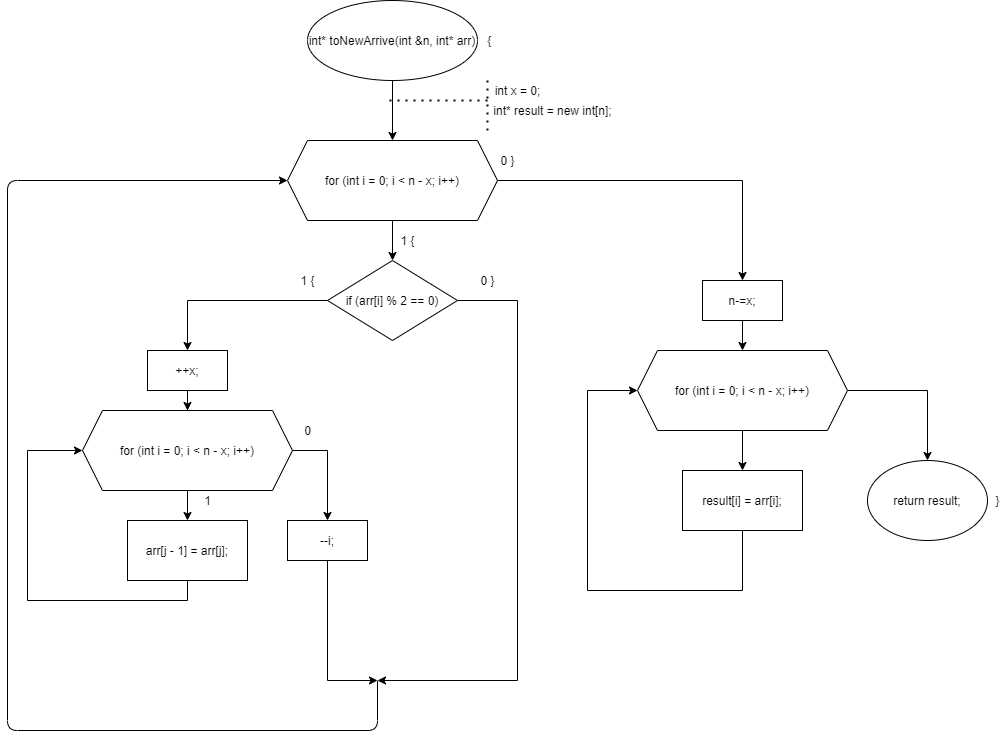
Функция, создающая массив слов:

Функция для строк:



Функция для четных позиций:

Функция для четных чисел:



Код программы

#include <iostream>

#include <windows.h>

using namespace std;

//создает массив слов

char\*\* createListWords(char\* row, int& count\_words)

{

char\*\* words = new char\* [255];

for (int i = 0; i < 255; i++)

{

words[i] = new char[255];

}

char pre\_letter = ' ';

char letter;

int i = 0, j = 0;

while (\*row != '\0')

{

letter = \*row;

if (letter == ' ' && pre\_letter != ' ')

{

words[i][j] = '\0';

j = 0;

i++;

}

else if ((letter != ' ' || pre\_letter != ' '))

{

words[i][j] = letter;

++j;

}

++row;

pre\_letter = letter;

}

words[i][j] = '\0';

count\_words = i + 1;

return words;

}

//удаляет массив слов

void deleteListWords(char\*\* words)

{

for (int i = 0; i < 255; i++)

{

delete[] words[i];

}

delete[] words;

}

//убирает четные слова из массива

char\* toNewArrive(char\* arr)

{

int count\_words = 0;

char\*\* words = createListWords(arr, count\_words);

char\* result = new char[255];

int count\_letter = 0;

for (int i = 0; i < count\_words; i += 2)

{

for (int j = 0; words[i][j] != '\0'; j += 1)

{

result[count\_letter] = words[i][j];

count\_letter += 1;

}

result[count\_letter] = ' ';

count\_letter += 1;

}

result[count\_letter] = '\0';

deleteListWords(words);

return result;

}

//убирает четные индексы из массива

int\* toNewArrive(int\* arr, int &n)

{

int\* result = new int[(n + 1) / 2];

int j = 0;

for (int i = 0; i < n; i += 2)

{

result[j] = arr[i];

j += 1;

}

n = (n + 1) / 2;

return result;

}

//убирает четные числа из массива

int\* toNewArrive(int &n, int\* arr)

{

int x = 0;

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

if (arr[i] % 2 == 0)

{

x++;

for (int j = i + 1; j < n; j++)

arr[j - 1] = arr[j];

i--;

}

n -= x;

int\* result = new int[n];

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

{

result[i] = arr[i];

}

return result;

}

int main()

{

bool flag;

setlocale(LC\_ALL, "Rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Что будем вводить: строка/числа? (1, 0)" << endl;

cin >> flag;

if (flag)

{

char row[255];

cout << "Введите слова через пробел: " << endl;

cin.get();

cin.getline(row, 255);

char\* result = toNewArrive(row);

cout << result << endl;

delete[] result;

}

else

{

int n;

cout << "Введите количество чисел: " << endl;

cin >> n;

int\* arr = new int[n];

int\* new\_arr = nullptr;

cout << "Введите " << n << " чисел: ";

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

{

cin >> arr[i];

}

cout << "Убрать четные индексы/числа? (1; 0): ";

cin >> flag;

if (flag)

{

new\_arr = toNewArrive(arr, n);

}

else {

new\_arr = toNewArrive(n, arr);

}

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

{

cout << new\_arr[i] << " ";

}

cout << endl;

delete[] arr;

delete[] new\_arr;

}

system("pause");

return 0;

}

Скриншоты программы:

