

Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем

Кафедра №12 «Компьютерные системы и технологии»



ОТЧЕТ

**О выполнении лабораторной работы №1
«Алгоритмизация обработки целых чисел»**

Студент: Гатченко А.С.

Группа: Б22-525

Преподаватель: Половнева Ю. А.

Москва — 2022

1. Формулировка индивидуального задания

Вариант №3. Дано целое число. Получить новое целое число только из разных цифр введенного числа. Например: из числа 23241 необходимо получить число 2341.

2. Описание использованных типов данных

При выполнении данной лабораторной работы использовался встроенный тип данных `int`, предназначенный для работы с целыми числами.

3. Описание использованного алгоритма

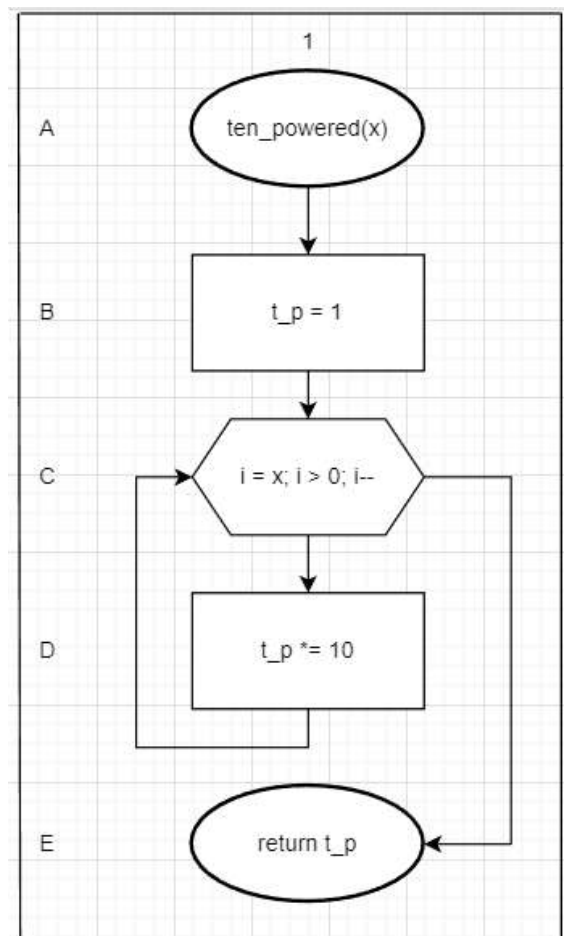
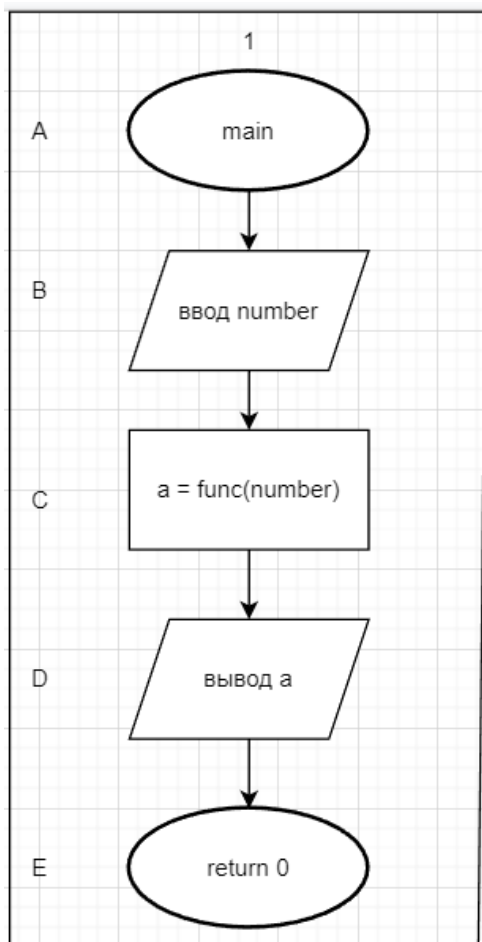


Рис. 1: Блок-схема алгоритма работы функции `main()`

Рис. 2: Блок-схема алгоритма работы функции `ten_powered()`

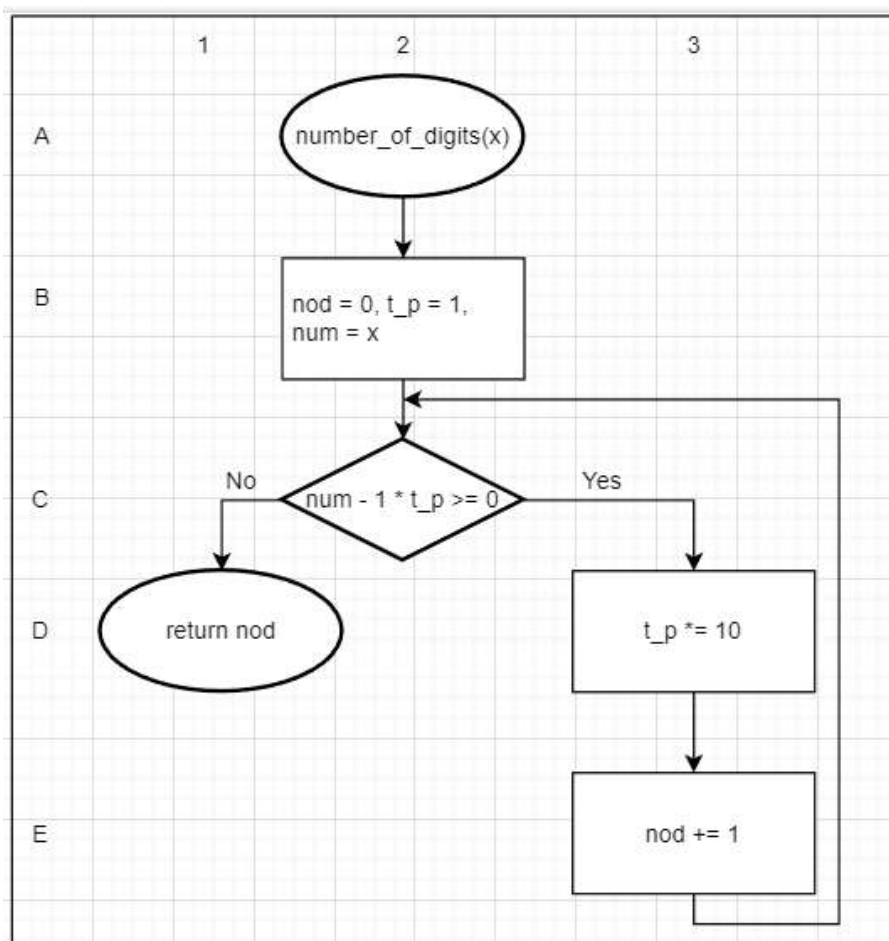


Рис. 3: Блок-схема алгоритма работы функции `number_of_digits()`

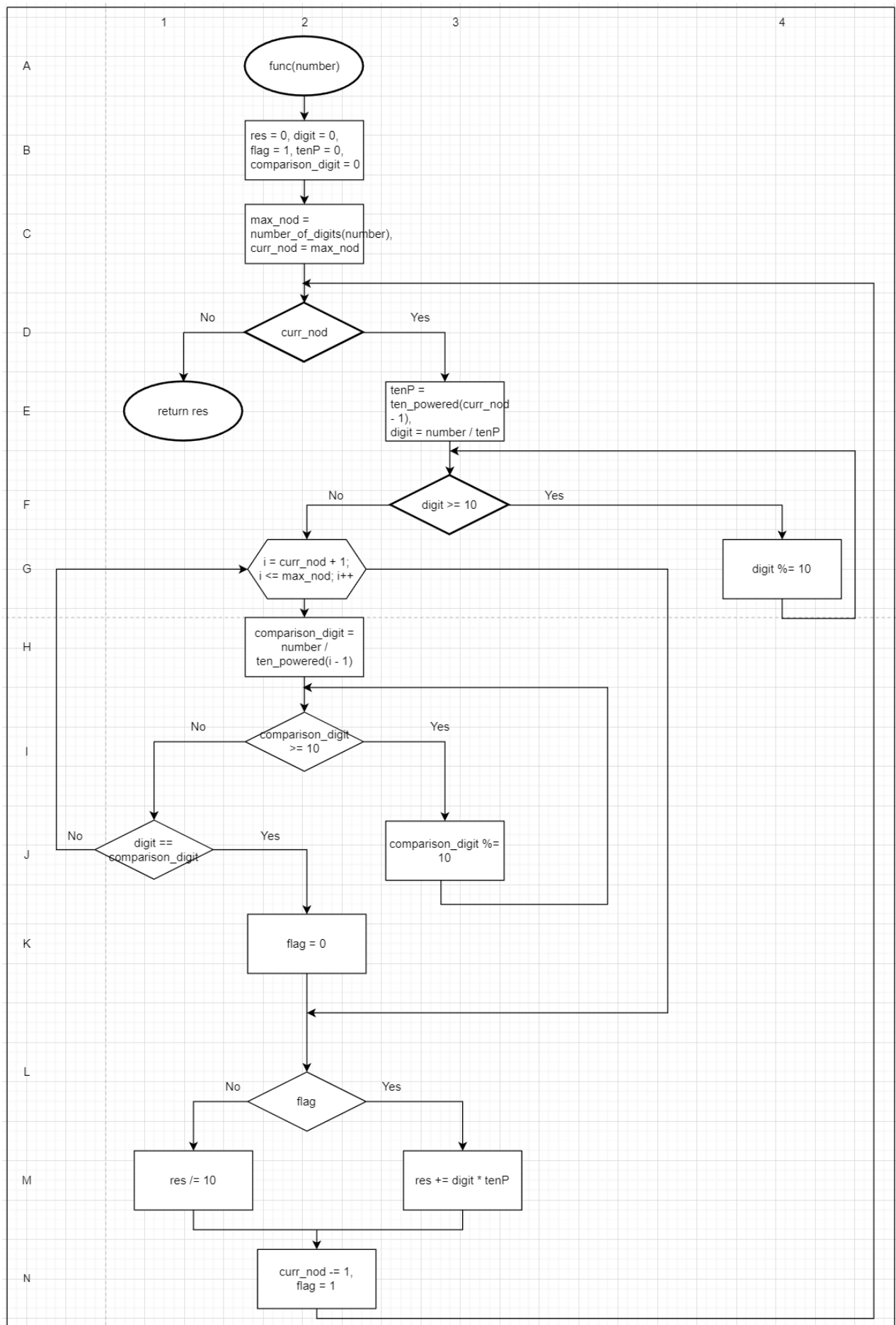


Рис. 4: Блок-схема алгоритма работы функции `number_of_digits()`

4. Исходные коды разработанных программ

Листинг 1: Исходные коды программы 15.09 (файл: 15.09.c)

```
#include <stdio.h>

int ten_powered(int number){
    int t_i_p = 1;
    for(int i = number; i > 0; i--){
        t_i_p *= 10;
    }
    return t_i_p;
}

int number_of_digits(int number){
    int nod = 0, t_p = 1, num = number;

    while (num - 1 * t_p >= 0){
        t_p *= 10,
        nod += 1;
    }
    return nod;
}

int func(int number){
    int tenP = 0, res = 0, digit = 0, flag = 1;
    int comparison_digit = 0, n_f = 0;
    int max_nod = 0, curr_nod = 0;

    if (number < 0){
        number = -number,
        n_f = 1;
    }

    max_nod = number_of_digits(number),
    curr_nod = max_nod;

    while (curr_nod){
        tenP = ten_powered(curr_nod - 1),
        digit = number / tenP;

        while (digit >= 10){
            digit %= 10;
        }

        for (int i = curr_nod + 1; i <= max_nod; i++){
            comparison_digit = number / ten_powered(i-1);

            while (comparison_digit >= 10){
                comparison_digit %= 10;
            }

            if (digit == comparison_digit){
                flag = 0;
                break;
            }
        }
    }
}
```

```

        if (flag){
            res += digit * tenP;
        }
        else{
            res /= 10;
        }

        curr_nod -= 1,
        flag = 1;
    }
    if (n_f){
        res = -res;
    }

    return res;
}

int main(){
    int number = 0;
    printf("Input your number:\n");
    scanf("%d", &number);
    printf("Your new number: %d\n", func(number));
    return 0;
}

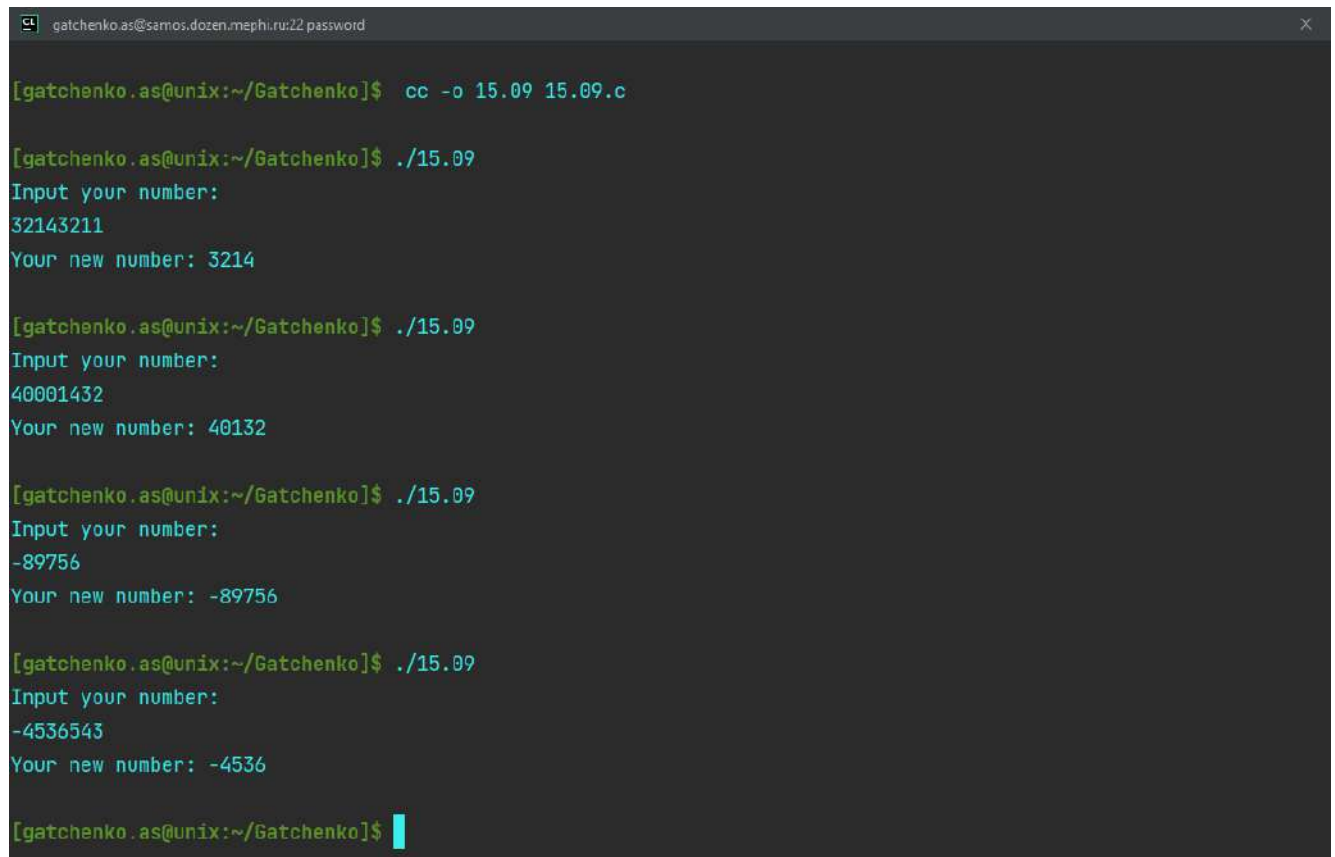
```

5. Описание тестовых примеров

Таблица 1: Тестовые примеры

Значение number	Ожидаемый вывод	Полученный вывод
32143211	3214	3214
40001432	40132	40132
-89756	-89756	-89756
-4536543	-4536	-4536

6. Скриншоты



```
gatchenko.as@samos.dozen.mephi.ru:22 password

[gatchenko.as@unix:~/Gatchenko]$ cc -o 15.09 15.09.c

[gatchenko.as@unix:~/Gatchenko]$ ./15.09
Input your number:
32143211
Your new number: 3214

[gatchenko.as@unix:~/Gatchenko]$ ./15.09
Input your number:
40001432
Your new number: 40132

[gatchenko.as@unix:~/Gatchenko]$ ./15.09
Input your number:
-89756
Your new number: -89756

[gatchenko.as@unix:~/Gatchenko]$ ./15.09
Input your number:
-4536543
Your new number: -4536

[gatchenko.as@unix:~/Gatchenko]$
```

Рис. 5: Сборка и запуск программы 15.09

7. Выводы

В ходе выполнения данной работы на примере программы, выполняющей получение нового числа без повторений цифр, были рассмотрены базовые принципы работы построения программ на языке C и обработки целых чисел:

1. Организация ввода/вывода.
2. Разработка функций.
3. Объявление и использование переменных.
4. Выполнение простейших арифметических операций над целочисленными операндами.
5. Использование циклов и условий.