Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчёт по лабораторной работе № 1**

Дисциплина: Низкоуровневое программирование

Тема: Машина Тьюринга-Поста

Вариант №12

Выполнила

студентка гр. 3530901/10003 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Р. Саушкина

(подпись)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А. Коренев

(подпись)

“ ” 2022 г.

Санкт-Петербург

2022

Оглавление

Оглавление

[Техническое задание 3](#_Toc115079111)

[Метод решения 4](#_Toc115079112)

[Описание состояний 5](#_Toc115079113)

[Работа программы 7](#_Toc115079114)

# Техническое задание

Написать программу преобразования унитарного кода в двоичный.

# Метод решения

Требование к исходным данным нет.

Рассмотрим пример преобразования унитарного числа 01000000 в двоичный.

Все незначащие нули заменяем пробелами, единицу заменяем на Х, чтобы было ограничение при дальнейших вычислениях:  
Х000000  
«Делим» на 2 получившееся «число» (заметим, что, если заменить все нули на единицы, то получим запись десятичного числа в унарном представлении, не делаем это для удобства вычислений), заменяя каждый второй 0 на пробел, если число было четным, то после деления пишем С, если нечетным, то убираем один 0 и пишем D:

Х0 0 0 С

Собираем все нули слева от символов C и D, повторяем операцию деления до тех пор, пока нули не закончатся:

Х СDD

Убираем Х и перед получившимся числом ставим А, чтобы было ограничение при дальнейших вычислениях:

АСDD

Заменяем последнюю букву на цифру (D=1, C=0), относительно нее переворачиваем число:  
А 110  
Убираем вспомогательный символ А:

110

Ответ готов (110).

# Описание состояний

Алфавит: 01АХСD

А и Х – вспомогательные символы, которые помогают отделить область работы программы от бесконечной ленты.

С и D – вспомогательные символы для записи двоичного числа, чтобы машина не путала число, которое преобразовывается с результатом (D=1, C=0).

Изначально головка находится на первой цифре числа (0 или 1 в зависимости от преобразуемого числа).

Q1 – отбрасывание незначащих нулей, замена единицы на Х, переход в состояние Q2, при нахождении пробела – пишем 0 и переходим в состояние Q4.

Q2 – выполнение деления числа на 2, если 0, то переход в состояние Q3, если число поделилось нацело, то в первом пробеле после числа записываем С и переходим в состояние Q4, если встречаются символы C и D, то пропускаем их и остаемся в состоянии Q2.

Q3 – продолжение деления, если 0, то заменяем его на пробел и переходим обратно в состояние Q2, если число было нечетным, то в первом пробеле записываем D и переходим в состояние Q10, если встречаются символы C и D, то пропускаем их и остаемся в состоянии Q2.

Q4 – выполняется поиск 0, при его нахождении заменятся на пробел и выполняется переход в состояние Q5, если встречаются символы C и D, то пропускаем их и остаемся в состоянии Q4, если встречается символ Х, то переходим в состояние Q6.

Q5 – пропускаем все пробелы, при нахождении 0, C или D переходим в состояние Q1.

Q6 – поиск первого 0 для продолжения деления, переход в состояние Q2, если встречаются символы C и D, то пропускаем их и переходим в состоянии Q9.

Q7 – заменяем один 0 на пробел (так как после деления нечетного числа на 2 остается остаток) и переходим в состояние Q4, вспомогательные символы C и D пропускаются, пробел заменяется вспомогательным символом А.

Q8 – вспомогательные символы C и D пропускаются, при нахождении пробела переходим в состояние Q9.

Q9 – осуществляется стирание вспомогательного символа Х, переход в 10 состояние.

Q10 – пропускаем пробелы, при нахождении вспомогательного символа C или D переходим в состояние Q7.

Q11 – поиск последнего вспомогательного символа C или D, переход в состояние Q12.

Q12 – замена последнего вспомогательного символа C или D на 0 или 1, соответственно (нужно для зеркального отображения числа), поиск начала полученного результата, остановка работы программы при нахождении 0 или 1.

Q13 – если обнаружен символ С, то на его место пишется Х, переход в состояние Q14, если обнаружен символ D, то на его место пишется Х, переход в состояние Q15, если обнаружен символ А, то он заменяется на пробел, переход в состояние Q12.

Q14 – пропуск всех символов, при нахождении пробела, осуществляется его замена на 0 и переход в состояние Q13.

Q15 – пропуск всех символов, при нахождении пробела, осуществляется его замена на 1 и переход в состояние Q13.

# Работа программы

 





