**Федеральное агентство связи ордена Трудового Красного Знамени федеральное государственное бюджетное общеобразовательное учреждение высшего образования «Московский технический университет связи и информатики»**

Кафедра МКиИТ

Лабораторная работа №6

Выполнили:

Студенты

группы БВТ1702

Перевозова Ю.

Зарудин И.

Романов В.

Москва 2019

**Задание:** Найти кратчайшее расстояние между двумя вершинами в графе. Найти все возможные пути между этими двумя вершинами в графе не пеpесекающиеся по

а) pебpам

б) веpшинам.

**Код программы:**

|  |
| --- |
| import Data.List  --дефолт : [[(2,3),(3,5)], [(4,5)], [(4,1)]]  --Элемент списка, финальный список  --На вход: [(2,3),(3,5)] [[]]  -- на выход: [ [(2,3)], [(3,5)] ]  create\_arr :: [(Int,Int)] -> [[(Int,Int)]] ->[[(Int,Int)]]  create\_arr [] full\_list = tail full\_list  create\_arr (x:xs) empty\_list = create\_arr xs (empty\_list ++ ((x:[]):[]))  --Получаем список [ [(2,3)], [(3,5)] ], пустой список для ответа [[(]], элемент [(4,5)] и номер узла - 2  add\_to\_list :: [[(Int,Int)]] -> [[(Int,Int)]] -> [(Int,Int)] -> Int -> [[(Int,Int)]]  add\_to\_list [] arr \_ \_ = tail arr  add\_to\_list (x:xs) arr elem i =  if (fst (head x) == i) then  add\_to\_list xs (arr ++ ((x ++ elem):[])) elem i  else  add\_to\_list xs (arr ++ (x : [])) elem i  --Полный список для графа [[(2,3),(3,5)], [(4,5)], [(4,1)]], пустой для ответа, номер итерации  func\_all\_path :: [[(Int,Int)]] -> [[(Int,Int)]] -> Int -> [[(Int,Int)]]  func\_all\_path [] ans \_ = ans  func\_all\_path (x:xs) ans i =  if (length x /= 1) then func\_all\_path xs (create\_arr x ans) (i+1)  else  func\_all\_path xs (add\_to\_list ans [[]] x i) (i+1)  --Подготовленный список, список для минимума, минимальная длина  fst\_min :: [[(Int,Int)]]->[(Int,Int)]->Int->[(Int,Int)]  fst\_min [] ans \_ = ans  fst\_min (x:xs) arr min = if ((snd\_min x 0) < min) then fst\_min xs x (snd\_min x 0)  else fst\_min xs arr min  snd\_min :: [(Int,Int)] -> Int -> Int  snd\_min [] sum = sum  snd\_min (x:xs) sum = snd\_min xs (sum + (snd x))  write\_path::[(Int,Int)]-> Int ->IO()  write\_path [] \_ = putStrLn ""  write\_path (x:xs) i = do  putStr "Вершина №"  putStr (show i)  putStr " - "  putStr (show (fst x))  putStr "; "  write\_path xs (i+1)  main :: IO()  main = do  putStrLn "Введите граф:"  str\_arr <- getLine  let arr = (read str\_arr) :: [[(Int,Int)]]  let ans = fst\_min (func\_all\_path arr [[]] 1) [] 100  putStrLn ""  putStr "Вершина №1 - 1; "  write\_path ans 2  putStr "Его длина равна: "  putStrLn (show (snd\_min ans 0))  putStrLn "" |

**Реализация программы на примерах:**

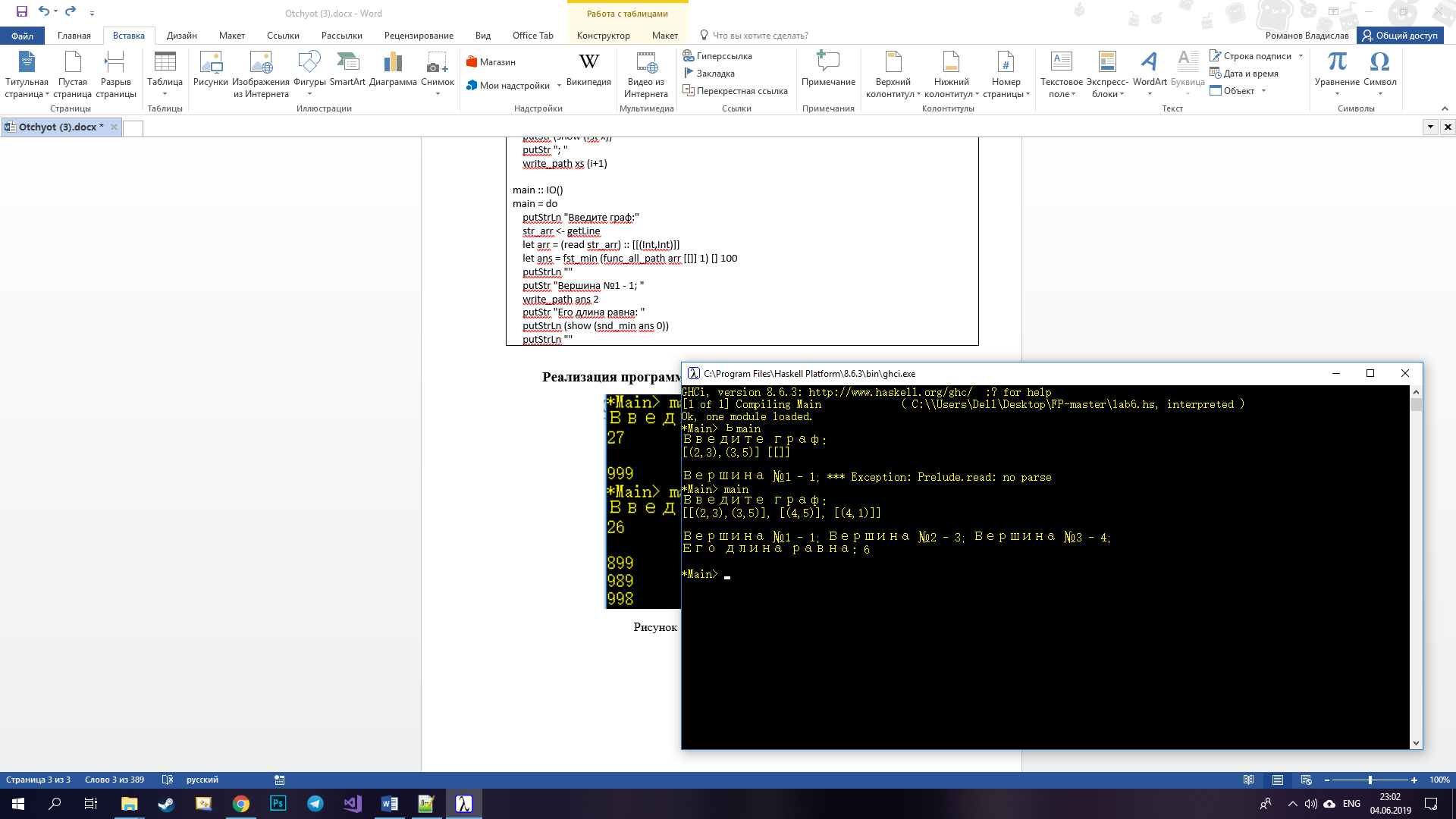


Рисунок 1 - Скриншот работы программы