«Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Лодейщикова В.В.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Отчёт

Лабораторной работе №1

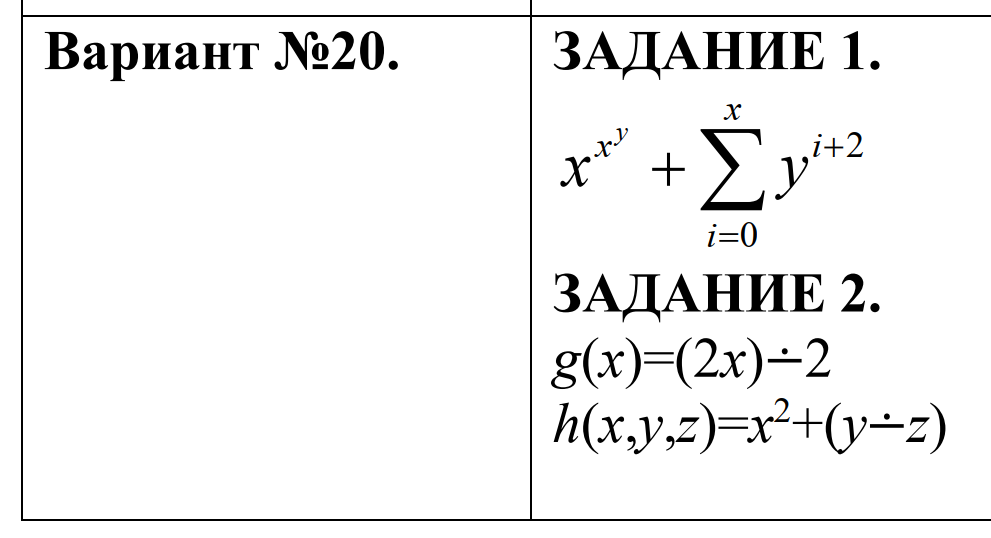
«Доказательство примитивной рекурсивности функции»

Студент группы ПИ 92 В.М. Шульпов

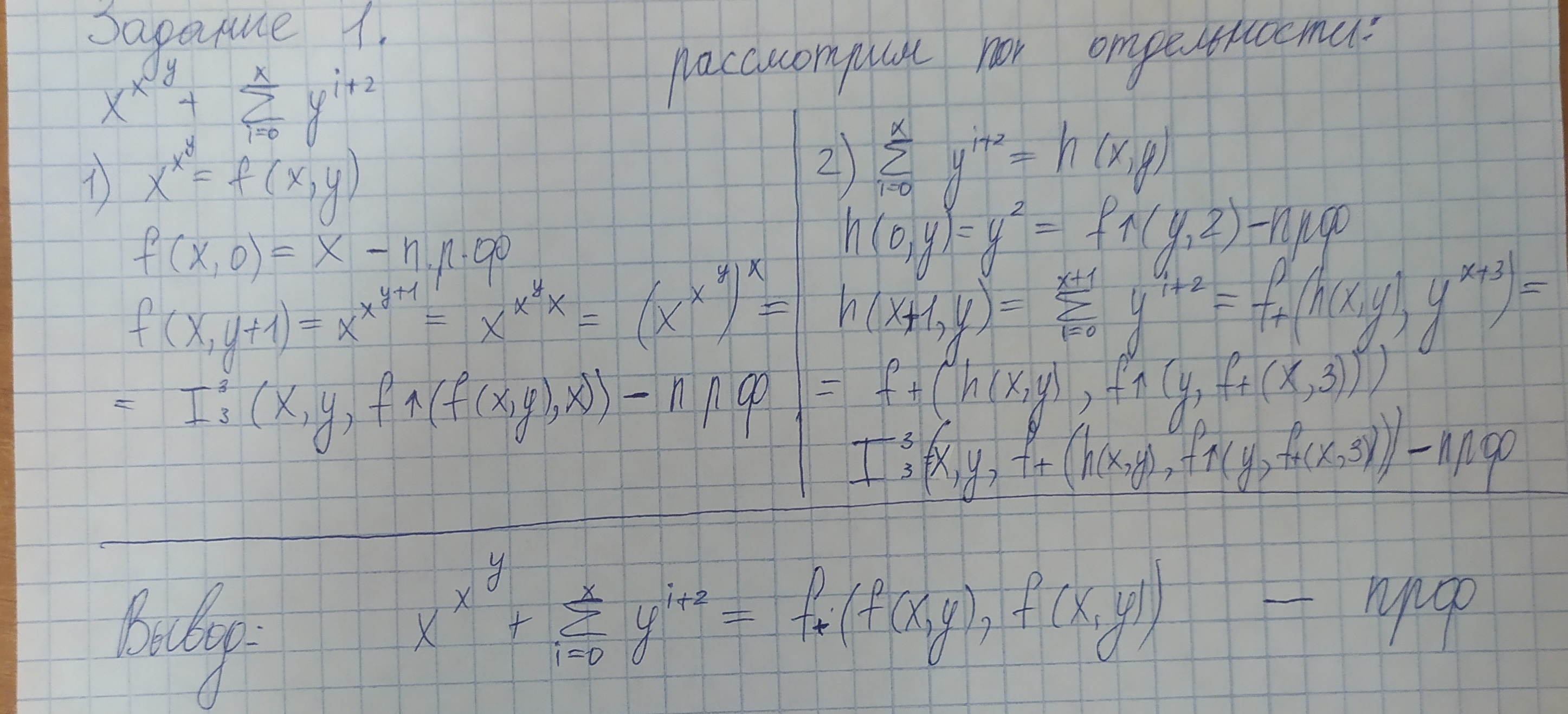
Преподаватель доцент, к.т. н. Лодейщикова В.В.

Барнаул 2021

Цели и задачи работы: изучение способов доказательства примитивной  
рекурсивности, разработка простейшей рекурсивной программы.  
Методика выполнения работы:• Изучить способы доказательства примитивной рекурсивности  
функций.  
• Доказать примитивную рекурсивность функции с использованием  
оператора суперпозиции и примитивной рекурсии (ЗАДАНИЕ 1).  
• Написать рекурсивную и нерекурсивную программы вычисления  
значения функции f , полученной оператором примитивной рекурсии  
R над функциями g и h (ЗАДАНИЕ 2).  
• Отладить и протестировать программы.  
• Продемонстрировать преподавателю работу программ.  
Требования к отчету:Отчет по лабораторной должен содержать титульный лист, доказательства  
примитивной рекурсивности функций, код рекурсивной и нерекурсивной  
программ, контрольные примеры.  
Защита лабораторной, помимо теоретических вопросов, включает выполнение  
практического задания по доказательству примитивной рекурсивности  
функции, предложенной преподавателем.



**Задание 1**



**Задание 2**

Код:

*# Лабортаторная работа №1 (Мат. логика)  
# Если g(x) = (2 \* x) ∸ 2, h(x, y, z) = (x \* x) + (y ∸ z), то  
# f(x, 0) = g(x) = (2 \* x) ∸ 2  
# f(x, y + 1) = h(x, y, f(x, y)) = (x \* x) + (y ∸ f(x, y))  
# Вычисление значений:  
# f(x, y) = h(x, y - 1, f(x, y - 1)) = (x \* x) + ((y - 1) ∸ f(x, y - 1))  
# f(x, 0) = (2 \* x) ∸ 2  
# f(x, 1) = h(x, 0, f(x, 0)) = (x \* x) + (0 ∸ [(2 \* x) ∸ 2]) = x \* x  
# f(x, 2) = h(x, 1, f(x, 1)) = (x \* x) + (1 ∸ [x \* x])  
# f(x, 3) = h(x, 2, f(x, 2)) = (x \* x) + (2 ∸ [(x \* x) + (1 ∸ [x \* x])])  
  
# Тесты  
# f(0, 0) = (0 \* 0) ∸ 2 = 0  
# f(1, 0) = (1 \* 1) ∸ 2 = 0  
# f(2, 0) = (2 \* 2) ∸ 2 = 2  
# f(3, 0) = (3 \* 3) ∸ 2 = 4  
# f(4, 0) = (4 \* 4) ∸ 2 = 6 и т.д.  
# ...  
# f(0, 1) = 0 \* 0 = 0  
# ...  
# f(0, 2) = (0 \* 0) + (1 ∸ [0 \* 0]) = 1  
# ...  
# f(0, 3) = (0 \* 0) + (2 ∸ [(0 \* 0) + (1 ∸ [0 \* 0])]) = 1  
# ...***from** colorama **import** Fore, Back  
  
  
**def** truncated\_dif(x, y):  
 *""" усеченная разность """* **return** x-y **if** x >= y **else** 0  
  
  
**def** f(x, y):  
 *""" функция рекурсивного алгоритма """* **if** y == 0:  
 **return** truncated\_dif(2 \* x, 2)  
 **return** (x \* x) + (truncated\_dif(y - 1, f(x, y - 1)))  
  
  
**def** f2(x, y):  
 *""" функция нерекурсивного алгоритма """* q = truncated\_dif(2\*x, 2)  
 **for** i **in** range(0, y):  
 q = (x \* x) + (truncated\_dif(i, f(x, i))) *# h(x, i, q)* **return** q  
  
  
**def** main():  
 max\_x = 5  
 max\_y = 5  
 print(Back.BLUE, Fore.LIGHTYELLOW\_EX + **"Рекурсивный алгоритм"** + **"\tНерекурсивный алгоритм"** + Back.LIGHTWHITE\_EX, Fore.BLACK)  
  
 **for** y **in** range(0, max\_y):  
 **for** x **in** range(0, max\_x):  
 print(**f'f({**x**}, {**y**}) = '**, f(x, y), end=**''**)  
 print(**f'\t\t\tf({**x**}, {**y**}) = '**, f2(x, y))  
  
  
main()

**Тесты:**

