

## Домашнее задание №2.

### Реализация и анализ примитивно-рекурсивной функции.

#### Вариант 2.9

Выполнил: Кравченко Александр Андреевич  
КТб01-6

#### 1. Вариант задания:

Найти функцию  $f(x_1, x_2)$ , полученную из функций  $g(x)$  и  $h(f, y, z)$  по схеме примитивной рекурсии:

$$\begin{cases} g(x) = x \\ h(f, y, z) = z \times y \end{cases}$$

#### 2. Схема рекурсии:

$$\begin{cases} f(0, x_2) = g(x_2) = x_2 \\ f(y + 1, x_2) = h(y, f(y, x_2), x_2) = x_2 \times f(y, x_2) \end{cases}$$

#### 3. Вывод аналитической записи функции:

Пусть  $x_2$  – переменная, вычислим функцию для нескольких значений  $x_1$ :

$$\begin{aligned} f(0, x_2) &= g(x_2) = x_2 \\ f(1, x_2) &= h(0, f(0, x_2), x_2) = h(0, x_2, x_2) = x_2^2 \\ f(2, x_2) &= h(1, f(1, x_2), x_2) = h(1, x_2^2, x_2) = x_2^3 \\ f(3, x_2) &= h(2, f(2, x_2), x_2) = h(2, x_2^3, x_2) = x_2^4 \\ f(4, x_2) &= h(3, f(3, x_2), x_2) = h(3, x_2^4, x_2) = x_2^5 \end{aligned}$$

Видим закономерность: в каждой формуле присутствует лишь одна переменная –  $x$ , а её степень всегда равна  $y + 1$ . Следовательно, аналитическая запись функции:

$$f(x_1, x_2) = x_2^{x_1+1}$$

#### 4. Проверка на двух примерах:

Проверим аналитическую запись функции на двух примерах:

1. Проверим для  $x_1 = 3, x_2 = 2$ :

$$\begin{aligned} f(0, 2) &= 2 \\ f(1, 2) &= h(0, f(0, 2), 2) = h(0, 2, 2) = 4 \\ f(2, 2) &= h(1, f(1, 2), 2) = h(1, 4, 2) = 8 \\ f(3, 2) &= h(2, f(2, 2), 2) = h(2, 8, 2) = 16 \end{aligned}$$

По построенной формуле:

$$f(3, 2) = x_2^{x_1+1} = 2^4 = 16$$

2. Проверим для  $x_1 = 5, x_2 = 4$ :

$$f(0, 4) = 4$$

$$\begin{aligned}
f(1,4) &= h(0, f(0,4), 4) = h(0,4,4) = 16 \\
f(2,4) &= h(1, f(1,4), 4) = h(1,16,4) = 64 \\
f(3,4) &= h(2, f(2,4), 4) = h(2,64,4) = 256 \\
f(4,4) &= h(3, f(3,4), 4) = h(3,256,4) = 1024 \\
f(5,4) &= h(4, f(4,4), 4) = h(4,1024,4) = 4096
\end{aligned}$$

По построенной формуле:

$$f(5,4) = x_2^{x_1+1} = 4^6 = 4096$$

## 5. Скриншоты выполнения программы на примерах из п. 4:

1.

```

Введите у: 3
Введите х: 2
f(0,2) = g(2) = 2
f(1,2) = h(0,f(0,2),2) = h(0,2,2) = 4
f(2,2) = h(1,f(1,2),2) = h(1,4,2) = 8
f(3,2) = h(2,f(2,2),2) = h(2,8,2) = 16
Вывод ПРФ: 16

```

Рисунок 1

2.

```

Введите у: 5
Введите х: 4
f(0,4) = g(4) = 4
f(1,4) = h(0,f(0,4),4) = h(0,4,4) = 16
f(2,4) = h(1,f(1,4),4) = h(1,16,4) = 64
f(3,4) = h(2,f(2,4),4) = h(2,64,4) = 256
f(4,4) = h(3,f(3,4),4) = h(3,256,4) = 1024
f(5,4) = h(4,f(4,4),4) = h(4,1024,4) = 4096
Вывод ПРФ: 4096

```

Рисунок 2

## 6. Листинг программы:

```

//ЮФУ, ИКТИБ, МОП ЭВМ
//Программирование и основы теории алгоритмов
//Д32 – Реализация и анализ примитивно-рекурсивной функции.
//КТ601-6, Кравченко Александр Андреевич
//11.04.2024

#include <iostream>
#include <conio.h>
#include <math.h>
#include <limits>

typedef unsigned long long ull;

// Рекурсивная функция PrimRecursive
// Принимает на вход два числа – х, у – переменные рекурсии
// Реализует рекурсивное вычисление функции
// На каждом шаге рекурсии выводит на экран строку, иллюстрирующую вычисление
// Возвращает одно число – результат работы рекурсии
ull PrimRecursive(ull, ull);

int main() {

```

```

setlocale(LC_ALL, "Russian");

ull x, y, res;
char cont;
bool limited;
std::cout << "Схема рекурсии:\nf(0,x) = g(x) = x\nf(y+1,x) = h(y,f(y,x),x) =
x*f(y,x)\n\n";

do {
    limited = 0;
    std::cout << "Введите y: ";
    std::cin >> y;

    std::cout << "Введите x: ";
    std::cin >> x;

    if (pow(x, y + 1) < ULLONG_MAX) {
        res = PrimRecursive(x, y);
        std::cout << "Вывод ПФ: " << res;

        std::cout << "\n\nНажмите любую клавишу, чтобы продолжить\nДля выхода
нажмите клавишу ESC на клавиатуре";
        cont = _getch();
        std::cout << "\n\n";
    }
    else {
        std::cout << "\nРезультат ПФ от введенных вами чисел выходит за пределы
области значений функции.\nПопробуйте ввести другие переменные.\n";
    }
} while (cont != 27 || limited);

exit(0);
}

ull PrimRecursive(ull x, ull y) {
    if (y == 0) {
        std::cout << "f(0," << x << ") = g(" << x << ") = " << x << "\n";
        return x;
    }
    ull curr = PrimRecursive(x, y - 1);
    ull next = pow(x, y + 1);
    std::cout << "f(" << y << "," << x << ") = h(" << y - 1 << "," << "f(" << y - 1
<< "," << x << "), " << x << ") = h(" << y - 1 << "," << curr << "," << x << ") = "
<< next << '\n';
    return next;
}

```