Nós,

Bruno Severo Camilo 41781619

Luiz Fernando Tagliaferro Brito 31861806

Declaramos que,

todas as respostas são fruto de nosso próprio trabalho,

não copiamos respostas de colegas externos a dupla,

não disponibilizamos nossas respostas para colegas externos a dupla e

não realizamos quaisquer outras atividades desonestas para nos beneficiar ou prejudicar

outros.

1)

a) (λ x y.x + y)

(\ x -> (\y -> (x + y)))10 20

=> 30

b)(λ x.x 2)

(\x -> x \* x)10

=> 100

c)(λ x.(λ y. x\*y))

(\ x -> (\y -> (x \* y)))10 20

=> 200

d)(λ x.(λ y. x²))

(\ x -> (\y -> (x ^ y)))10 20

=> 100000000000000000000

2)

tripla x y z

| x^2 + y^2 == z^2 = "e uma tripla de Pitagoras"

| y^2 + z^2 == x^2 = "e uma tripla de Pitagoras"

| z^2 + x^2 == y^2 = "e uma tripla de Pitagoras"

| otherwise = "Nao e tripla de Pitagoras"

3)

duracao inicio fim

|inicio < fim = fim-inicio

|inicio > fim = (fim+24) - inicio

|inicio == fim = 24

|otherwise = -1

4)

Criterio de parada x = 1 e retorno y

Se x é par, retorno afundando na recursao(nao acresta y na soma)

Caso contrario, retorno a funçao com o valor somado de y

multRussa x y

| (x == 1) = y

| (mod x 2 == 0) = multRussa (div x 2) y \* 2

| otherwise = ( y + multRussa (div x 2) y \* 2)

5)

Se a funçao numerospares receber um array vazio, retorna zero -> metodo de parada

Verifico se o primeiro elemento mod 2 da resto zero, se sim, chamo a funçao

novamente com o proximo elemento do array e +1 pois encontrei um elemento valido(par)

Caso nao satisfaça essa condiçao, chamo novamente a funçao sem o +1

numerosPares [] = 0

numerosPares (primeiro\_elemento:resto\_lista)

|(mod primeiro\_elemento 2 == 0) = 1 + numerosPares resto\_lista

|otherwise = 0 + numerosPares resto\_lista

6)

retorno uma lista com os valores menores ou iguais a m

verficaçao x € lista | x <= m

elementosMenores lista m = [x | x <- lista, x <= m]