

A feladat rövid összefoglalása

System T kifejezések (természetes számok és magasabbrendű függvények nyelve) kiértékelése és α -egyenlőségének eldöntése. Szintaxis: jegyzetben 7.1 definíciók alapján, viszont a lambdák nál nem tüntetjük fel a kötött változók típusát. Operációs szemantika: 7.3. ABT-k α -egyenlősége: 2.2. A feladat megoldásához nincs szükség a típusozási szabályokra.

A program bemenete két zárt e_1 és e_2 kifejezés. Ha léteznek e_1' és e_2' értékek (val az operációs szemantika szerint), amelyekre igaz, hogy e_1 és e_2 nulla vagy több lépésben rendre e_1' -re és e_2' -re értékelődnek, és e_1' és e_2' α -ekvivalensek, akkor a kimenet YES, egyébként NO.

Kevésbé formálisan, a feladat a két input kifejezés kiértékelése, majd a kapott két érték α -egyenlőségének vizsgálata.

Bemeneti formátum

A bemenet JSON formátumban van megadva. Minden input szintaktikailag helyes és típusozható (de a típus nem releváns a feladathoz). Minden sorra a standard inputról egy sort kell a standard outputra küldeni.

```
e1 = suc zero
e2 = (λ n. λ m. rec m (x. suc x) n) zero (suc zero)

e1 = (λ f. λ x. f x) (λ x. x) zero
e2 = rec zero (x. suc (suc zero)) zero

e1 = λ x. x
e2 = λ y. y

e1 = λ x. rec zero (x.suc x) x
e2 = λ y. rec zero (y.suc y) y

e1 = rec (λ x. x) (x. λ y. x (x y)) (suc zero)
e2 = λ y. rec (λ z. z) (x. λ y. x (x y)) zero (rec (λ q. q) (x. λ z. x (x z)) zero y)

e1 = rec (λ x. x) (x. λ y. x (x y)) zero
e2 = λ x. zero

e1 = λ x. rec x (y. x) zero
e2 = λ x. x
```

```
{
  "e1": {
    "op": "Suc",
    "args": {
      "op": "Zero",
      "args": []
    },
    "e2": {
      "op": "App",
      "args": {
        "op": "App",
        "args": {
          "op": "Lam",
          "args": [
            "f",
            {
              "op": "Lam",
              "args": [
                "x",
                {
                  "op": "App",
                  "args": {
                    "op": "Var",
                    "args": "m"
                  },
                  "x",
                  {
                    "op": "Suc",
                    "args": {
                      "op": "Var",
                      "args": "x"
                    },
                    {
                      "op": "Var",
                      "args": "n"
                    }
                  ]
                },
                {
                  "op": "Zero",
                  "args": []
                }
              ]
            }
          ]
        }
      }
    }
  },
  "e1": {
    "op": "App",
    "args": {
      "op": "App",
      "args": {
        "op": "Lam",
        "args": [
          "f",
          {
            "op": "Lam",
            "args": [
              "x",
              {
                "op": "App",
                "args": {
                  "op": "Var",
                  "args": "f"
                },
                {
                  "op": "Var",
                  "args": "x"
                }
              ]
            }
          ]
        }
      },
      {
        "op": "Lam",
        "args": [
          "x",
          {
            "op": "Var",
            "args": "x"
          }
        ]
      },
      {
        "op": "Zero",
        "args": []
      }
    },
    "e2": {
      "op": "Rec",
      "args": {
        "op": "Zero",
        "args": [
          [],
          "x",
          {
            "op": "Suc",
            "args": {
              "op": "Suc",
            "args": {
              "op": "Zero",
              "args": []
            }
          }
        ]
      }
    }
  },
  "e1": {
    "op": "Lam",
    "args": [
      "x",
      {
        "op": "Var",
        "args": "x"
      }
    ],
    "e2": {
      "op": "Lam",
      "args": [
        "y",
        {
          "op": "Var",
          "args": "y"
        }
      ]
    }
  },
  "e1": {
    "op": "Lam",
    "args": [
      "x",
      {
        "op": "Rec",
        "args": {
          "op": "Zero",
          "args": []
        },
        "x",
        {
          "op": "Suc",
          "args": {
            "op": "Var",
            "args": "x"
          }
        }
      ]
    },
    "e2": {
      "op": "Lam",
      "args": [
        "y",
        {
          "op": "Rec",
          "args": {
            "op": "Zero",
            "args": [
              [],
              "y",
              {
                "op": "Suc",
                "args": {
                  "op": "Var",
                  "args": "y"
                },
                {
                  "op": "Var",
                  "args": "y"
                }
              ]
            }
          }
        ]
      }
    }
  },
  "e1": {
    "op": "Rec",
    "args": {
      "op": "Lam",
      "args": [
        "x",
        {
          "op": "Var",
          "args": "x"
        }
      ],
      "x",
      {
        "op": "Lam",
        "args": [
          "y",
          {
            "op": "App",
            "args": {
              "op": "Var",
              "args": "x"
            },
            {
              "op": "App",
              "args": {
                "op": "Var",
                "args": "x"
              },
              {
                "op": "Var",
                "args": "y"
              }
            ]
          }
        ],
        {
          "op": "Suc",
          "args": {
            "op": "Zero",
            "args": []
          }
        }
      ]
    },
    "e2": {
      "op": "Lam",
      "args": [
        "y",
        {
          "op": "App",
          "args": {
            "op": "Rec",
            "args": {
              "op": "Lam",
              "args": [
                "z",
                {
                  "op": "Var",
                  "args": "z"
                }
              ],
              "x",
              {
                "op": "Lam",
                "args": [
                  "y",
                  {
                    "op": "App",
                    "args": {
                      "op": "Var",
                      "args": "x"
                    },
                    {
                      "op": "App",
                      "args": {
                        "op": "Var",
                        "args": "x"
                      },
                      {
                        "op": "Var",
                        "args": "z"
                      }
                    ]
                  }
                ]
              }
            }
          }
        ]
      }
    }
  },
  "e1": {
    "op": "Rec",
    "args": {
      "op": "Lam",
      "args": [
        "x",
        {
          "op": "Var",
          "args": "x"
        }
      ],
      "x",
      {
        "op": "Lam",
        "args": [
          "y",
          {
            "op": "App",
            "args": {
              "op": "Var",
              "args": "x"
            },
            {
              "op": "App",
              "args": {
                "op": "Var",
                "args": "x"
              },
              {
                "op": "Var",
                "args": "y"
              }
            ]
          }
        ],
        {
          "op": "Zero",
          "args": []
        }
      ]
    },
    "e2": {
      "op": "Lam",
      "args": [
        "x",
        {
          "op": "Zero",
          "args": [
            [],
            {
              "op": "App",
              "args": {
                "op": "Var",
                "args": "x"
              },
              {
                "op": "Var",
                "args": "x"
              }
            }
          ]
        }
      ]
    }
  },
  "e1": {
    "op": "Lam",
    "args": [
      "x",
      {
        "op": "Rec",
        "args": {
          "op": "Var",
          "args": "x"
        },
        "y",
        {
          "op": "Var",
          "args": "x"
        }
      ]
    },
    "e2": {
      "op": "Lam",
      "args": [
        "x",
        {
          "op": "Var",
          "args": "x"
        }
      ]
    }
  }
}
```

YES
YES
YES
YES
YES
NO
NO

A JSON formátum általánosan:

```
<var> ::= <JSON string>
```

```
<exp> ::=
```

```
    { "op": "Var",  "args": <var> }  
  | { "op": "App",  "args": [<exp>, <exp>] }  
  | { "op": "Lam",  "args": [<var>, <exp>] }  
  | { "op": "Rec",  "args": [<exp>, <var>, <exp>, <exp>] }  
  | { "op": "Zero"  "args": [] }  
  | { "op": "Suc",  "args": <exp> }
```

```
<input> ::= {"e1": <exp>, "e2" : <exp>}
```

Implementációs részletek

Java

Java nyelv esetében két lehetőség adott. Az első esetben a beadott megoldás a gyökérben tartalmazhat egy `Main.java` állományt, ebben kell lennie egy `Main` osztálynak, ahol a `main()` metódus elérhető. A `Main` osztály ne legyen benne semmilyen `package`-ben. A programnak ilyenkor tehát a következőképpen fordíthatónak kell lennie:

```
$ javac Main.java
```

valamint a következőképpen futtathatónak:

```
$ echo '{"con": {"x":"Str"}, "exp": {"op":"Var","args":"x"}, "ty": "Str"}' | java Main
```

```
YES
```

```
$
```

Külső programkönyvtárakat `jar` kiterjesztésű fájlként ugyanabban a könyvtárban kell megadni, mint ahol a `Main.java` van.

Feltételezhetjük, hogy a tesztelésnél a JDK 1.7, 1.8 és a Maven 3.1 változata áll rendelkezésre.