Avishai Yaniv: 307916023
Alex Abramov: 314129438

PPL – ASSIGNMENT 1

---------- Part 1 ----------

1.

Definition:
An expression is called *Referential Transparent* if all functions involved in it are pure functions, meaning that they return same value for the same arguments and evaluation has no side-effects. Impure functions can be also included in *Referential Transparent* expressions if their values are discarded and they have no side-effects as well.

An advantage:
By its definition, *Referential Transparent* eliminates side-effects.

An example:

```
function add(x:number,y:number) : number{ return x+y;}
function sub(x:number,y:number) : number{ return x-y;}
add(sub(1,2), 3);    // => ans: 2
add(-1, 3);          // => ans: 2
```

As we can see, sub(1,3) = -1 , therefore, the first expression can be replaced with the second without changing the value of the expression.

2.

```
function ans(employees) : number{
    let tmp = employees.filter(x => x.salary > 9000),
    sum : number = tmp.reduce((sum,employee) => sum + employee.salary,0);
    return sum/tmp.length;
}
```

3.

- [{name: String, degrees: [{name: String, years: Number}]}]

- (f(y:T1) => T2 , g(z:T3) => T1 , h(n: Number) => T3) => ((x: Number) => T2)

- (pred: (x:T1) = > boolean, arr: T1[]) => boolean

- (f: (x:T1) => Number, a:T1[]) => Number