

Assignment 1

מגישים : בר פרלמן 305026882

מורן נפטון 208474544

חלק 1 :

1. **Referential transparency-** is a property of functional programs which makes the inductive process of evaluation of expressions (which only focuses on the structure of the input program) easier. It means that the value of a program (called an *expression* in FP) depends only on its sub-expressions, and that if you substitute a sub-expression in an expression by another expression that is equivalent, then the resulting expression is equivalent to the original, without changing the program's behavior. This requires that the expression is pure, that is to say the expression value must be the same for the same inputs and its evaluation must have no side effects.

An advantage : if one evaluates an expression twice, one obtains the same result. The relative order in which one evaluates (non over lapping) sub-expressions of a program makes no difference to the value of the program. *This property enables optimization methods such as parallel evaluation of sub-expressions to speed up code.*

An example: $\sin(x)$ is referentially transparent, since it will always give the same result for each particular x .

2. The Converting of the function :

```
const averageSalaryOver9000 = (employees) => {  
  const result = employees.filter(e=>e.salary>=9000);  
  return (result.reduce((acc,curr)=> acc+curr.salary,0))  
    /(result.reduce((acc,curr)=>acc+1,0));  
}
```

3.

1.

```
interface title{
    name : string;
    years : number;
}

interface Student {
    name : string;
    degrees : title[];
}
```

```
let studentsArr : Student[] = [{name: 'Sara', degrees: [{name: 'CS',
years: 3},{name: 'Biology', years: 4}]}];
```

2.

$(f:(z:T3) \Rightarrow T4), g:(y:T2) \Rightarrow T3, h:(x:\text{number}) \Rightarrow T2) \Rightarrow ((x:\text{number}) \Rightarrow T4)$

Note that x is of type number because the addition with the operand 1 is performed in h.

3.

$(\text{pred}:T1 \Rightarrow \text{boolean}, \text{arr}: T1[]) \Rightarrow \text{boolean}$

explanation:

let arr values to be of type T1.

Acc in reduce is initialized to the boolean false.

Cur has to be boolean for the defined operation.

The returned value of the reduce is the acc itself so its boolean, thus:

$(\text{reduce}: (\text{acc}:\text{boolean}, \text{cur}:\text{boolean}) \Rightarrow \text{boolean})$

so the definition of $\text{arr.map}(\text{pred}):\text{boolean}[]$ is for the reduce operation to be legal.

4.

$(f : T1 \Rightarrow \text{number}, a : T1[]) \Rightarrow \text{number}$

explanation:

let a b array of values of type T1.

For validity of activating f on a f get T1 and returns T2 which is a number for the validity of the reduce definition.