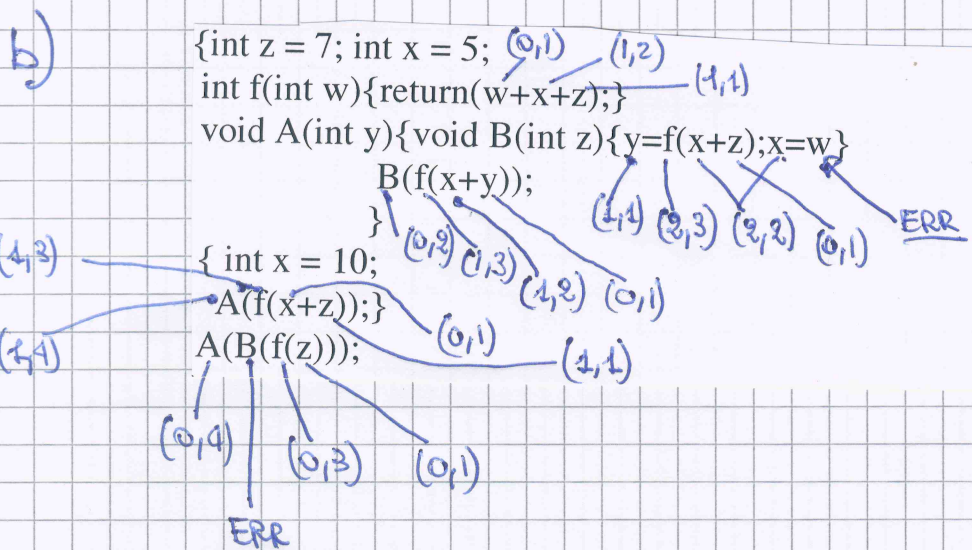
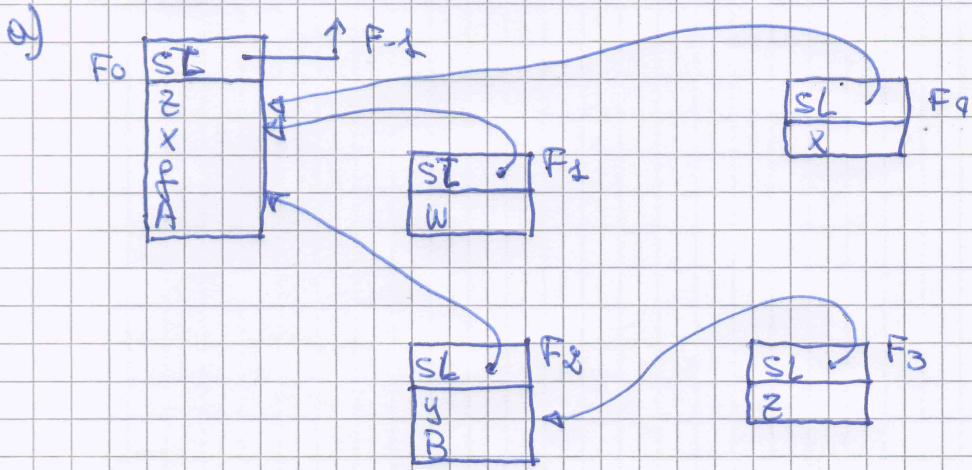


EX 1. (Apples 3 - 16 pages - Perceptron. PoCupine)



EX 2. (Apples 3 - 16 pages - Perceptron. PoCupine)

(a)  $= v$   
 $= (\text{foldR } f \ v \ lR)$

(b)  $\text{foldR } (||) \ \text{false} \ (\text{list.map } (\text{foldR } (||) \ \text{true}) \ lR)$

(c) let rec foldRT = fun v l e → match l with

  [] → v

  x :: lR → foldRT f v lR (fun u → z (f x u))

---

Ex3. (Appello3 - 16 giugno - Paradigmi. Soluzione)

```
public class RelazioneA<A>{

    public RelazioneA () {
        size = 0;
    }

    public boolean isIn (A x, A y){
        return (size==0 || getDue(x).contains(y));
    }

    public boolean isTransitive() {
        RelazioneA<A> cursor = this;
        LinkedList<A> empty = new LinkedList<A>();
        while (!(cursor.size==0) && isClosed(cursor.uno,empty))
            cursor = cursor.next;
        return (cursor.size==0);
    }

    private boolean isClosed(A x, LinkedList<A> cons) {
        /* true sse (x in cons) ||
           (x,y),(y,z)in R(this) implica (x,z)in R(this) */
        boolean ret = true;
        if (!(cons.contains(x))) {
            LinkedList<A> lx = getDue(x);
            for (A y : lx){
                if (!(x.equals(y))){
                    cons.add(x);
                    ret = isClosed(y,cons);
                    LinkedList<A> ly = getDue(y);
                    for (A z : ly) ret = ret && isIn(x,z);
                }
            };
        };
        return ret;
    }
}
```