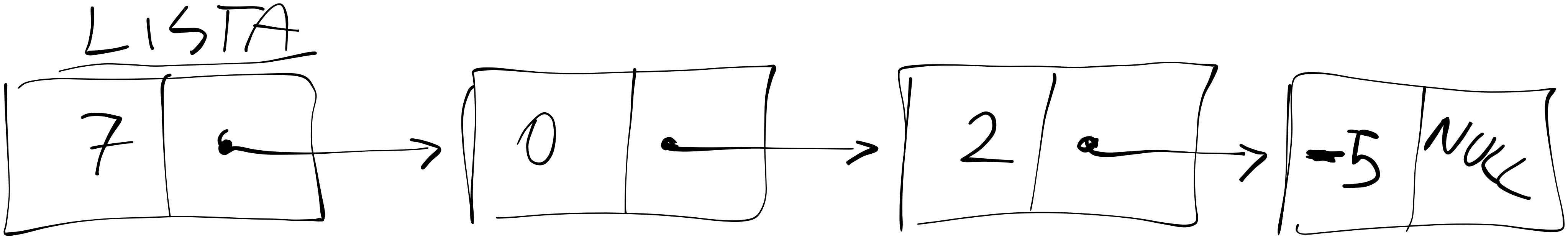
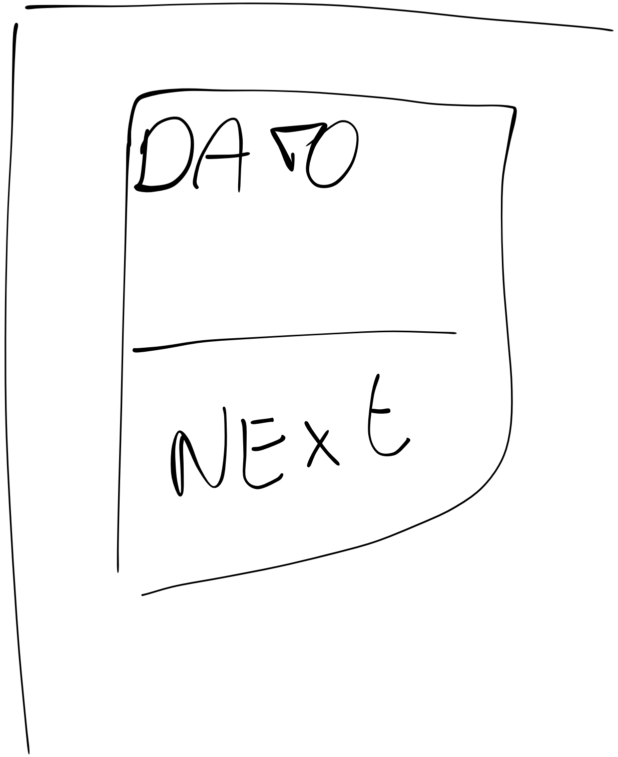


RETURN :

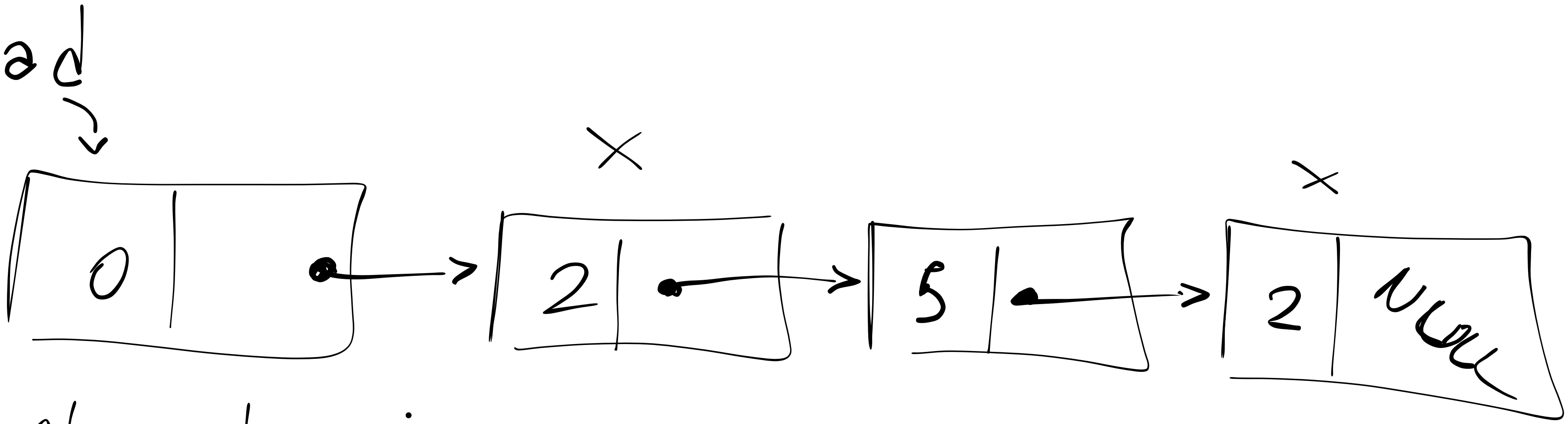
9 ← SOMMA DI SOLI VAL >0 .



```
int sommaPositivi(Nodo* head){
        INT SOMMA=0;
    Nodo * TMP=HEAD;

    WHILE(TMP != NULL){
        IF(TMP->dato>0){
            SOMMA += TMP->dato;
        }
        TMP=TMP->NEXT;
    }
    RETURN (SOMMA);
}
```

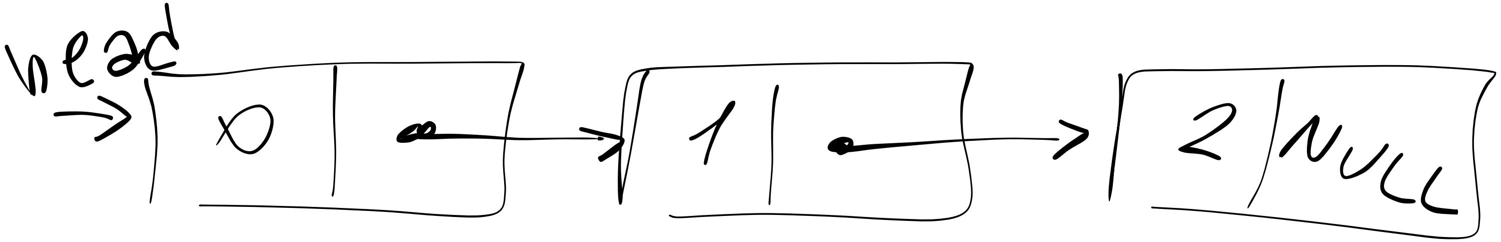
```
int somPosiRic(Nodo* head){
    IF(HEAD!=NULL){
        IF(HEAD->DAto >0){
            REturn head->Dato+SomPosiRic(hEAD
            ->NEXt);
        }
        REturN 0+somPosiRk(hEAD->NEXt);
    }
    REtURN 0; //CASO BASE (head==NULL)
}
```

int prodVal (Nodo* head, int val){



int val = 2;

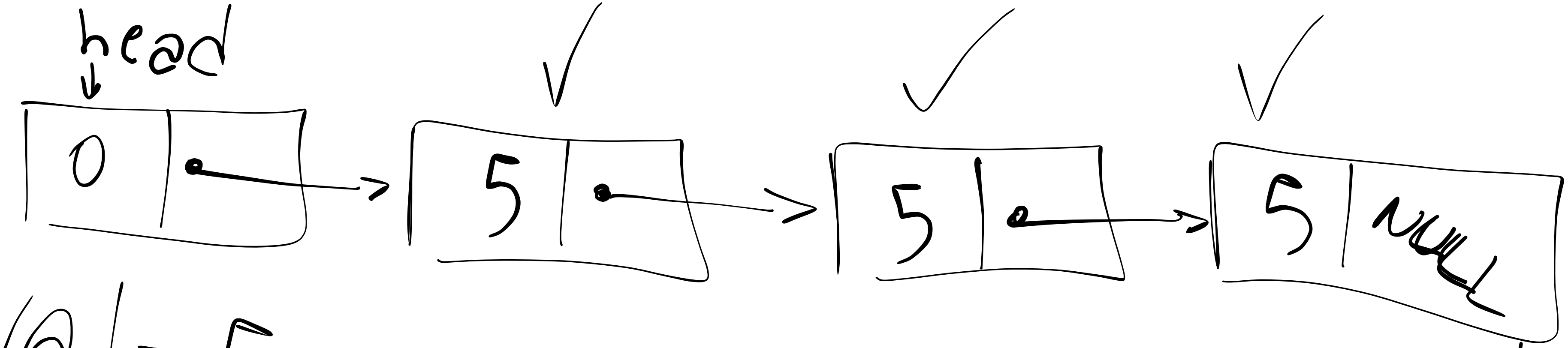
$\Rightarrow 2 \cdot 2 = 4$



val = 2;

=> 2

SE



val = 5

=> 5 · 5 · 5 = 125

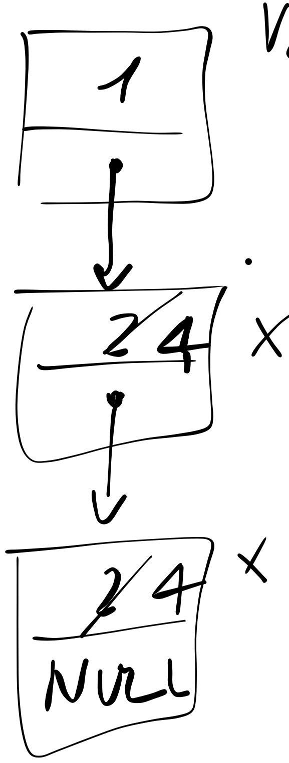
count = 3

$(val^{count})E$

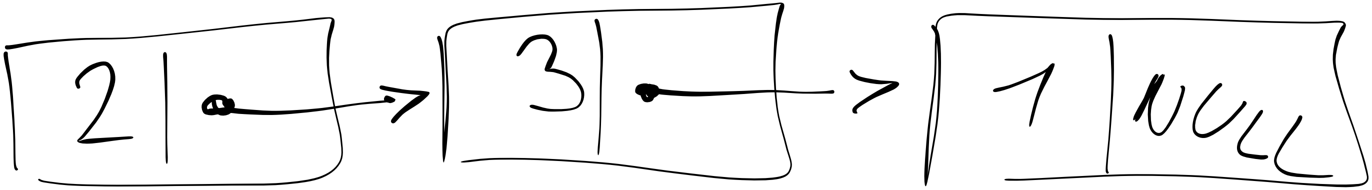
```
int prodVal (Nodo *head, int val){
    int prod = 1; √
  Nodo* tmp=head;
   while(tmp!=NULL){
      if(tmp->dato==val){

          prod =tmp->dato* prod;
      }
      tmp=tmp->Next;
   }
   return(prod);
}
```

==VAL

Val=2



prod =2·2=4



VAL = 5

⇒ ? → (0 || 1)

```
INT    PRODOTTOVAL(NODO * HEAD, INT VAL){
        INT MOLT=0;
        INT  FLAG=0;
        WHILE(HEAD!=NULL){
            IF(HEAD->DATO==VAL){
                FLAG++;
            }
            IF(HEAD->DATO==VAL && FLAG==1){
                MOLT=MOLT+HEAD->DATO;
            }
            ELSE IF(HEAD->DATO==VAL && FLAG>1){
                MOLT=MOLT*HEAD->DATO;
            }
            HEAD=HEAD->NEXT;
        }
        RETURN MOST;
    }
```

```
INT PRODVAL(NODO* HEAD, INT VAL){
        NODO* TMP = HEAD;
        INT CNT=0;
        WHILE(TMP){
            IF(TMP->DATO == VAL){
                CNT++;
            }
            TMP=TMP->NEXT;
        }
        IF(CNT>0){
            RETURN (INT) POW(VAL,CNT);
        }ELSE{
            RETURN 0;
        }
}
```