

#### Listing 4: schema de contrat

```

1 #include <limits.h>
2 /*@ axiomatic auxmath {
3   @ axiom rule1: \forall int n; n > 0 ==> n*n == (n-1)*(n-1)+2*n+1;
4   @ } */
5 /*@ requires 0 <= x;
6   ensures \result == x*x;
7 */
8 int power2(int x);
9
10 /*@ requires 0 <= x;
11   ensures \result == x*x;
12 */
13 int p(int x);
14
15 /*@ ensures \result == ???;
16 */
17
18 int check(int n){
19   int r1, r2, r;
20   r1 = power2(n);
21   r2 = p(n);
22   if (r1 == r2)
23     { r = ???;
24     }
25   else
26     { r = ???;
27     };
28   return r;
29 }

```

### Notations pour WP

La définition structurelle des transformateurs de prédicats est rappelée dans le tableau ci-dessous:

S	$wp(S)(P)$
$X := E(X, D)$	$P[e(x, d)/x]$
SKIP	P
$S_1; S_2$	$wp(S_1)(wp(S_2)(P))$
IF B S <sub>1</sub> ELSE S <sub>2</sub> FI	$(B \Rightarrow wp(S_1)(P)) \wedge (\neg B \Rightarrow wp(S_2)(P))$

### Axiomes et règles d'inférence de la Logique de Hoare

- Axiome d'affectation:  $\{P(e/x)\} X := E(X) \{P\}$ .
- Axiome du saut:  $\{P\} \text{skip} \{P\}$ .
- Règle de composition : Si  $\{P\} S_1 \{R\}$  et  $\{R\} S_2 \{Q\}$ , alors  $\{P\} S_1; S_2 \{Q\}$ .
- Si  $\{P \wedge B\} S_1 \{Q\}$  et  $\{P \wedge \neg B\} S_2 \{Q\}$ , alors  $\{P\} \text{if B then } S_1 \text{ then } S_2 \text{ fi} \{Q\}$ .
- Si  $\{P \wedge B\} S \{P\}$ , alors  $\{P\} \text{while B do S od} \{P \wedge \neg B\}$ .
- Règle de renforcement/affaiblissement: Si  $P' \Rightarrow P$ ,  $\{P\} S \{Q\}$ ,  $Q \Rightarrow Q'$ , alors  $\{P'\} S \{Q'\}$ .

Fin de l'énoncé