Considérons le programme ci-dessous. Celui-ci cherche à déterminer si un nombre x est premier en essayant tous les diviseurs possibles de 2 jusqu'à $\lfloor\sqrt{x}\rfloor$. Nous allons démontrer formellement que le programme est correct malgré l'usage risqué de nombres en virgule flottante.

```
bool est_premier(uint64_t x)
{
  if (x <= 1)
    return false;

  double   y = x;
  double   z = sqrt(y);
  uint64_t d = 2;

  while (d <= z) {
    if (x % d == 0)
      return false;

    d++;
  }

  return true;
}
```

Dans le reste du document, nous faisons ces hypothèses standards :

— « `sqrt` » est conforme à la norme IEEE 754 ;

— les valeurs de type `double` sont représentées en nombre en virgule flottante double précision binaire de la norme IEEE 754, avec le mode d'arrondi qui approxime au nombre le plus près, où les bris d'égalité se font vers le nombre dont le dernier bit est pair.

1 Borne sur la racine carrée

Rappelons, qu'étant donné $x \in \mathbb{R} \setminus \{0\}$, nous écrivons $\overline{x}$ afin de dénoter le nombre en virgule flottante normalisé arrondi au nombre le plus près de x, et où un bris d'égalité se fait vers le nombre dont le dernier bit est pair. Rappelons également que l'erreur relative de x est définie par $\mathrm{err}(x) := (x - \overline{x})/x$.

Rappelons que les `double` ont une mantisse de 53 bits et que leurs nombres normalisés appartiennent au domaine $\mathbb{R}' := \{x \in \mathbb{R} \setminus \{0\} : 2^{-1022} \leq |x| \leq 2^{1024} - 2^{971}\}$. La borne suivante sur l'erreur relative est bien connue.

Proposition 1. $|\mathrm{err}(x)| \leq 2^{-53}$ *pour tout* $x \in \mathbb{R}'$.

Celle-ci permet de faire l'observation ci-dessous.

Corollaire 2. *Soient* $n \in \mathbb{N}$ *et* $x \in \mathbb{R}' \cap [2^n, 2^{n+1})$. *Nous avons* $|x - \overline{x}| < 2^{n-52}$.

Démonstration. Par la proposition 1, nous avons $|(x - \overline{x})/x| \leq 2^{-53}$. Puisque $x \geq 1$, cela implique que $|x - \overline{x}| \leq x \cdot 2^{-53}$. Comme $x < 2^{n+1}$, on en conclue que $|x - \overline{x}| < 2^{n-52}$. □

Établissons maintenant une borne sur l'écart entre la racine carrée de x et la racine carrée de son approximation.

Proposition 3. *Soient $n \in \mathbb{N}$ et $x \in \mathbb{R}' \cap [2^n, 2^{n+1})$. Nous avons $|\sqrt{x} - \sqrt{\overline{x}}| < 2^{\lfloor n/2 \rfloor - 52}$.*

Démonstration. Nous avons

$$\begin{aligned}
\left|\sqrt{x} - \sqrt{\overline{x}}\right| &= \frac{|x - \overline{x}|}{\sqrt{x} + \sqrt{\overline{x}}} && \text{(car } (a-b)(a+b) = a^2 - b^2\text{)} \\
&< \frac{2^{n-52}}{\sqrt{x} + \sqrt{\overline{x}}} && \text{(par le corollaire 2)} \\
&\leq \frac{2^{n-52}}{\sqrt{x} + \sqrt{x - 2^{n-52}}} && \text{(car } \overline{x} \geq x - 2^{n-52} \text{ par le corollaire 2)} \\
&\leq \frac{2^{n-52}}{\sqrt{2^n} + \sqrt{2^n - 2^{n-52}}} && \text{(car } x \geq 2^n\text{)} \\
&\leq \frac{2^{n-52}}{\sqrt{2^n} + \sqrt{2^{n-1}}} && \text{(car } 2^n - 2^{n-52} \geq 2^{n-1}\text{)} \\
&\leq \frac{2^{n-52}}{2 \cdot \sqrt{2^{n-1}}} && \text{(car } 2^n \geq 2^{n-1}\text{)} \\
&= \frac{2^{n-52}}{2^{(n+1)/2}} \\
&\leq \frac{2^{n-52}}{2^{\lceil n/2 \rceil}} \\
&= 2^{\lfloor n/2 \rfloor - 52}.
\end{aligned}$$

□

Théorème 4. *Soit $n \in \mathbb{N}$ tel que $1 \leq \lfloor n/2 \rfloor \leq 51$ et soit $x \in \mathbb{R}' \cap [2^n, 2^{n+1})$. Nous avons $\overline{\sqrt{\overline{x}}} \geq \lfloor \sqrt{x} \rfloor$.*

Démonstration. Par définition, $\overline{x}$ se représente exactement en nombre en virgule flottante double précision. Ainsi, selon la norme IEEE 754, $\overline{\sqrt{\overline{x}}}$ est égal au nombre que l'on obtient en calculant d'abord $\sqrt{\overline{x}}$ avec précision infinie, puis en l'arrondissant.

Remarquons que $\sqrt{x} \geq \sqrt{2^n} = 2^{n/2} \geq 2^{\lfloor n/2 \rfloor}$ et $\sqrt{x} < \sqrt{2^{n+1}} = 2^{(n+1)/2} \leq 2^{\lfloor n/2 \rfloor + 1}$. Ainsi, en représentation en virgule flottante infinie (binaire), $\sqrt{x}$ peut s'écrire de cette forme :

$$1{,}d_1 d_2 \cdots \times 2^{\lfloor n/2 \rfloor}, \tag{1}$$

où $d_i = 0$ pour une infinité d'indices i.

Par la proposition 3, nous avons $|\sqrt{x} - \sqrt{\overline{x}}| < 2^{\lfloor n/2 \rfloor - 52}$. Ainsi, si $\sqrt{\overline{x}} - \sqrt{x}$ est négative, alors sa plus petite valeur possible peut s'écrire de cette forme en représentation en virgule flottante infinie (binaire) :

$$-1{,}c_1c_2\cdots \times 2^{\lfloor n/2 \rfloor - 53}, \tag{2}$$

où $c_i = 0$ pour une infinité d'indices i.

Mettons (1) et (2) sur un exposant commun :

$$\sqrt{x}: \qquad 1{,}d_1 \cdots d_{\lfloor n/2 \rfloor}\, d_{\lfloor n/2 \rfloor + 1} \cdots d_{52}\ d_{53}d_{54}\cdots \quad \times 2^{\lfloor n/2 \rfloor}, \tag{3}$$

$$\sqrt{\overline{x}} - \sqrt{x}: \qquad -0{,}\underbrace{00\cdots0000}_{\lfloor n/2 \rfloor \text{ fois}}\ \underbrace{000000\cdots0000}_{52 - \lfloor n/2 \rfloor \text{ fois}}\, c_1c_2\cdots\cdots \times 2^{\lfloor n/2 \rfloor}. \tag{4}$$

Notons que $\lfloor \sqrt{x} \rfloor = 1, d_1 \cdots d_{\lfloor n/2 \rfloor} \times 2^{\lfloor n/2 \rfloor}$. La somme de (3) et (4) est $\sqrt{\overline{x}}$. Si la soustraction sous-jacente n'emprunte aucun bit à $d_1 \cdots d_{\lfloor n/2 \rfloor}$, alors l'arrondi de $\sqrt{\overline{x}}$ est forcément supérieur ou égal à $\lfloor \sqrt{x} \rfloor$, et nous avons ainsi terminé. Supposons donc qu'il y a emprunt. Le résultat est de cette forme :

$$\sqrt{\overline{x}}: \qquad 1, d_1 \cdots d_i\ 0\ 1 \cdots 1\ 1 \cdots 1\ 1\ b_{54}b_{55}\cdots \times 2^{\lfloor n/2 \rfloor}.$$

Puisque le $52^{\text{ème}}$ bit après la virgule est impair, on arrondit forcément vers le haut (même si $b_{54} = b_{55} = \cdots = 0$) :

$$\overline{\sqrt{\overline{x}}}: \qquad 1, d_1 \cdots d_i\ 1\ 0 \cdots 0\ 0 \cdots 0 \times 2^{\lfloor n/2 \rfloor}.$$

Par conséquent, $\overline{\sqrt{\overline{x}}} = \lfloor \sqrt{x} \rfloor$. □

2 Le programme est correct

Nous faisons une dernière observation (bien connue), puis nous démontrons que le programme est correct.

Proposition 5. *Soit $x \in \mathbb{N}_{\geq 2}$ un nombre qui n'est pas premier. Il existe $d \in \mathbb{N}$ tel que $2 \leq d \leq \lfloor \sqrt{x} \rfloor$ et d divise x.*

Démonstration. Comme x n'est pas premier, il possède au moins deux facteurs premiers. Afin d'obtenir une contradiction, supposons que tous les facteurs premiers de x sont strictement supérieurs à $\lfloor \sqrt{x} \rfloor$. Soient $p_1, p_2, \ldots, p_k$ les facteurs premiers de x. Par hypothèse, nous avons $p_i > \lfloor \sqrt{x} \rfloor$ pour tout $i \in [1..k]$. Comme chaque p_i est entier, nous avons forcément $p_i > \sqrt{x}$ pour tout $i \in [1..k]$.

Ainsi $x = p_1p_2\cdots p_k \geq p_1p_2 > \sqrt{x} \cdot \sqrt{x} = x$, ce qui est une contradiction. □

Théorème 6. *Le programme est correct.*

Démonstration. Soit $x \in \mathbb{N}$ une entrée du programme. Comme x est un entier non signé de 64 bits, nous avons $0 \leq x < 2^{64}$. Si $x \in \{0, 1\}$, alors clairement le programme retourne le bon résultat. Supposons donc que $x \geq 2$. Par la proposition 5, l'algorithme utilisé par le programme est correct. En effet, si x est premier, alors il ne possède aucun diviseur, et si x est premier alors il possède au moins un diviseur d tel que $d \leq \lfloor\sqrt{x}\rfloor \leq \sqrt{x}$. Néanmoins, ce raisonnement suppose que toutes les opérations sont implémentées de façon exactes, ce qui n'est pas le cas. Il y a trois opérations risquées :

— « `double y = x;` » effectue une conversion qui approxime `x` ;

— « `double z = sqrt(y);` » calcule une approximation de la racine carrée de `y`, qui elle-même est une approximation de `x` ;

— « `d <= z` » compare un entier à un nombre en virgule flottante, bien que `d` soit possiblement trop grand pour être représenté exactement.

Montrons que ces trois points ne sont pas problématiques. Supposons que x n'est pas premier (car autrement il n'y a pas de problème). Nous considérons deux cas. Avant de procéder, remarquons que $\lfloor\sqrt{x}\rfloor$ est représentable exactement dans un `double` car

$$\lfloor\sqrt{x}\rfloor \leq \sqrt{x} < \sqrt{2^{64}} = 2^{32} \leq 2^{53}.$$

Cas 1 : $2 \leq x \leq 2^{53}$. Comme $x \leq 2^{53}$, x est représentable exactement dans un `double`. Autrement dit, $\overline{x} = x$. Par la proposition 5, le plus petit facteur premier p de x satisfait $p \leq \lfloor\sqrt{x}\rfloor$. Ainsi, p est représentable exactement dans un `double`.

Selon la norme IEEE 754, $\overline{\sqrt{\overline{x}}}$ est égal au nombre que l'on obtient en calculant d'abord $\sqrt{\overline{x}}$ avec précision infinie, puis en l'arrondissant. Nous avons $p \leq \lfloor\sqrt{x}\rfloor \leq \sqrt{x} = \sqrt{\overline{x}}$. Ainsi, $\sqrt{\overline{x}}$ n'est pas arrondi sous p. Plus formellement :

$$p \leq \overline{\sqrt{\overline{x}}} = \mathtt{z}.$$

Puisque tous les entiers $\mathtt{d} \in [2..p]$ sont représentables exactement en `double`, la condition « `d <= z` » est évaluée correctement jusqu'à ce qu'on identifie correctement que $\mathtt{d} = p$ est un diviseur.

Cas 2 : $2^{53} < x < 2^{64}$. Il existe un entier $n \in [53..63]$ tel que $2^n \leq x < 2^{n+1}$. Ainsi, nous avons $x \in \mathbb{R}'$ et clairement $1 \leq \lfloor n/2\rfloor \leq 51$. Par conséquent, nous pouvons invoquer le théorème 4. Nous avons donc

$$\mathtt{z} = \overline{\sqrt{\overline{x}}} \geq \lfloor\sqrt{x}\rfloor.$$

Ainsi, la boucle `while` considère tous les diviseurs `d` tels que $\mathtt{d} \leq \lfloor\sqrt{x}\rfloor$, pourvu que la comparaison « `d <= z` » soit correcte. Montrons que c'est le cas.

Comme mentionné précédemment, $\lfloor\sqrt{x}\rfloor$ est représentable exactement dans un `double`. Ainsi, toutes les comparaison sont exactes pour $\mathtt{d} \in [2..\lfloor\sqrt{x}\rfloor]$. □