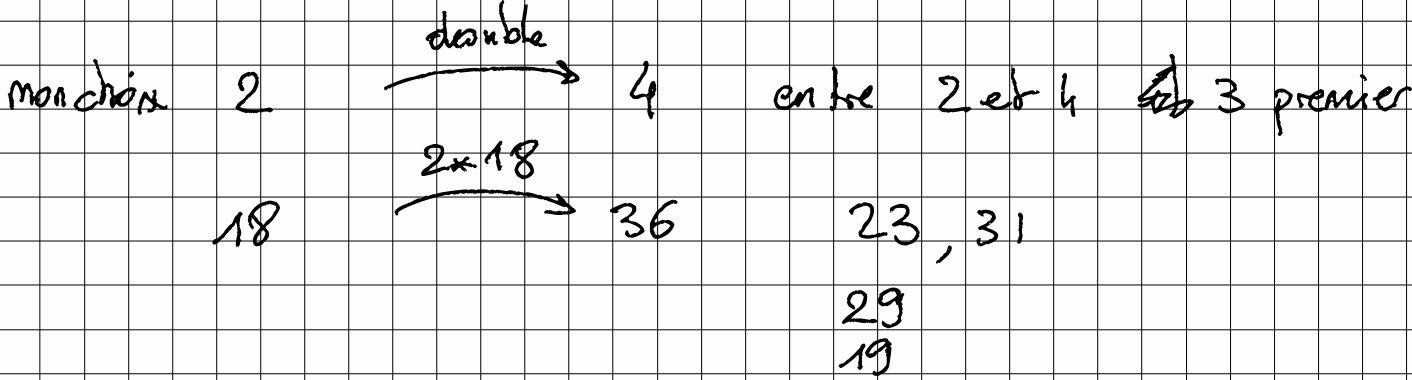


Le Postulat de Bertrand affirme qu'entre un entier et son double, il existe toujours un nombre premier.



soit il n'y a aucun élément premier \Rightarrow Postulat faux

18 premier NON
19 premier OUI

soit il y a effectivement un élément de l'intervalle qui est premier \Rightarrow Postulat est vrai

1. choisir un nombre N
 2. On calcule son double
3. on génère un intervalle en N et son double
4. Pour chacun des éléments de l'intervalle on "regarde" si il est premier

double, premier, construire un intervalle, parcourir l'intervalle

2*

facile quand c'est
entre 2 entiers

avec une autre
boucle
donné

itération

unsigned int n;
scanf("%d", &n);

premier?

intervalle

```
for (i = n; i <= 2 * n; i++) {  
    if (premier?(i)) {  
        on sort }  
}
```

// break //

$i++ \equiv i = i + 1$

on sort: → break

→ on modifie i [$i = 2 * n + 1$

pour l'amener à rendre fausse

l'expression $i \leq 2 * n$

→ return

→ drapeau flag
↳ nouvelle variable
f qui vaut 0
ou 1

Les booléens n'existent pas en C

↳ On se utilise les entiers pour "coder" les booléens

soit b un entier

si $b = 0$ on dira que b est faux

si $b \neq 0$ on dira que b est vrai