

# Algorithmique - bilan

## Informatique et Science du Numérique



# Algorithmique - bilan

Informatique et Science du Numérique



## Coding party

1. Écrire un programme qui élève un nombre au carré
2. Écrire un programme qui teste si un nombre est pair
3. Écrire un programme qui calcule la factorielle d'un nombre
4. Écrire un programme qui teste si un nombre est premier
5. Écrire un programme qui inverse un texte

# Algorithmique - bilan

## Informatique et Science du Numérique



### Python – v1.0

```
# saisie nombre premier ou pas...
nbr = input("Quel nombre voulez-vous tester ? ")

# initialisation
result = "premier"

for i in range(2, nbr): # on teste systématiquement TOUS les diviseurs
    if nbr % i == 0:
        result = "non premier"

#affichage
print(result)
```

# Algorithmique - bilan

## Informatique et Science du Numérique



### Python – v2.0

```
# saisie nombre premier ou pas...
nbr = input("Quel nombre voulez-vous tester ? ")

# initialisation
result = "premier"

if nbr % 2 == 1:          # on élimine les nombres pairs
    for i in range (3, nbr):
        if nbr % i == 0:
            result = "pas premier"
            break        # on sort dès qu'on trouve un diviseur
else:
    if nbr > 2:
        result = "pas premier"

print(result)
```

# Algorithmique - bilan

## Informatique et Science du Numérique



### break & continue

```
total = 0  
maxi = input("Donnez un intervalle pas trop grand")
```

```
# que fait-on ?  
for i in range (1, maxi):  
    if i % 2 == 0 or i % 3 == 0:  
        continue  
    if i % 101 == 0:  
        break  
    total += 1
```



- Structure de contrôle non algorithmique
- Sortie de boucle à posteriori

```
#affichage... mais quoi ?  
print(total)
```

# Algorithmique - bilan

## Informatique et Science du Numérique



### Python – v3.0

```
# saisie nombre premier ou pas...
nbr = input("Quel nombre voulez-vous tester ? ")

# initialisation
premier = True

if nbr % 2 == 1:           # on élimine les nombres pairs
    i = 3                 # on ne teste pas les diviseur pairs
    # on ne peut utiliser une boucle for
    while i < nbr and premier == True: # CE = CC ^ !CA
        if nbr % i == 0:
            premier = False # on sort dès qu'un diviseur est trouvé
        else:
            i += 2         # pas de diviseur pair
    else:
        if nbr > 2:
            premier = False
```

```
#affichage
if premier == True:
    print("premier")
else:
    print("pas premier")
```

# Algorithmique - bilan

## Informatique et Science du Numérique



### Python – v3.1

```
# saisie nombre premier ou pas...
nbr = input("Quel nombre voulez-vous tester ? ")

# initialisation
premier = True

if nbr % 2 == 1:          # on élimine les nombres pairs
    i = 3
    n = (int) sqrt(nbr) + 1 #optimisation
    while i < n and premier == True: # CE = CC ^ !CA
        if nbr % i == 0:
            premier = False
        else:
            i += 2        # on ne teste pas les diviseurs pairs
    else:
        if nbr > 2:
            premier = False
```

```
# affichage
if premier == True:
    print("nombre premier")
else:
    print("ce nombre n'est pas premier")
```

# Algorithmique - bilan

## Informatique et Science du Numérique



### Langage C

```
#include <stdio.h>
#include <math.h>

int main()
{
    // declaration
    int nbr, i = 3;
    unsigned char premier = 1;

    // test si un nbr est premier ou pas...
    printf("Quel nombre voulez-vous tester ? ");
    scanf("%d", &nbr);

    int n = (int) sqrt(nbr) + 1;
```

```
// traitement
if ( nbr > 2 && nbr % 2 == 0 ) // test nbr pair
    premier = 0 ;
else
    while ( premier && i < n )
        if ( nbr % i == 0 )
            premier = 0 ;
        else
            i += 2 ;

// affichage
if ( premier )
    printf("nombre premier");
else
    printf("ce nombre n'est pas premier");

return 0;
}
```



# Algorithmique - bilan

## Informatique et Science du Numérique



### Langage C++

```
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    int nbr, i(3); // declaration
    bool premier = true;

    // test si un nbr est premier ou pas...
    cout << "Quel nombre voulez-vous tester ? ";
    cin >> nbr;

    // traitement
    int n = static_cast<int>(sqrt(nbr)) + 1;

    // traitement
    if ( nbr > 2 && nbr % 2 == 0 ) // test nbr pair
        premier = false;
    else
        while ( premier && i < n )
            if ( nbr % i == 0 )
                premier = false;
            else
                i += 2;

    // affichage
    if ( premier )
        cout << "nombre premier" << endl;
    else
        cout << "ce nombre n'est pas premier" << endl;

    return 0;
}
```

# Algorithmique - bilan

Informatique et Science du Numérique



## Coding party++

1. Écrire une fonction qui teste si un nombre est premier
2. Écrire un programme qui indique le nombre total de nombres premiers dans un intervalle  $[1..n]$
3. Modifier le programme ci-dessus pour qu'il indique le temps  $t$  de traitement
4. Tracer le graphique  $t = f(n)$  pour un algorithme optimisé et non optimisé

# Algorithmique - bilan

## Informatique et Science du Numérique



### Python

```
import time
```

```
def est_premier(nbr):
```

```
    # fonction: teste si un nombre est premier
    # in:      nbr: nombre à tester
    # out:     vrai si nbr est premier, faux sinon
    premier = True
```

```
if nbr % 2 == 1: # on élimine les nombres pairs
```

```
    i = 3
```

```
    n = (int) sqrt(nbr) + 1
```

```
    while i < n and premier == True:
```

```
        if nbr % i == 0:
```

```
            premier = False
```

```
        else:
```

```
            i += 2 # pas de diviseurs pairs
```

```
else:
```

```
    if nbr > 2:
```

```
        premier = False
```

```
return premier
```

```
# cherche les nombres premiers dans un intervalle
nbr = input("Indiquez l'intervalle")
```

```
start = time.clock();
```

```
# traitement
```

```
total = 0
```

```
for i in range (1, nbr+1):
```

```
    if est_premier(i) == True:
```

```
        total += 1
```

```
end = time.clock()
```

```
# affichage
```

```
print "temps traitement : {0} ms".format((end - start) * 1000)
```

```
print "total : {0}".format(total)
```

# Algorithmique - bilan

## Informatique et Science du Numérique



### Langage C

```
#include <time.h>
```

```
unsigned char est_premier(const int nbr)
{
// function:   teste si un nombre est premier
// in:         nbr: nombre à tester
// out:        vrai si nbr est premier, faux sinon
```

```
    const int n = (int) sqrt(nbr) + 1;
```

```
    int i = 3;
    unsigned char premier = 1;
```

```
    if ( nbr > 2 && nbr % 2 == 0 ) // test nbr pair
        premier = 0;
    else
```

```
        while ( premier && i < n )
            if ( nbr % i == 0 )
                premier = 0;
            else
                i += 2;
```

```
    return premier;
}
```

```
int main()
{
// declaration
int nbr, total = 0;
```

```
// test si un nbr est premier ou pas...
printf("Donnez un intervalle ");
scanf("%d", &nbr);
```

```
    clock_t start = clock();
```

```
    for (int i=1; i <= nbr; i++)
        if ( est_premier(i) )
            total++;
```

```
    clock_t end = clock();
```

```
    printf("temps traitement : %.2Lf ms\n", (long double) (end - start) / CLOCKS_PER_SEC * 1000);
    printf("total : %d", total);
```

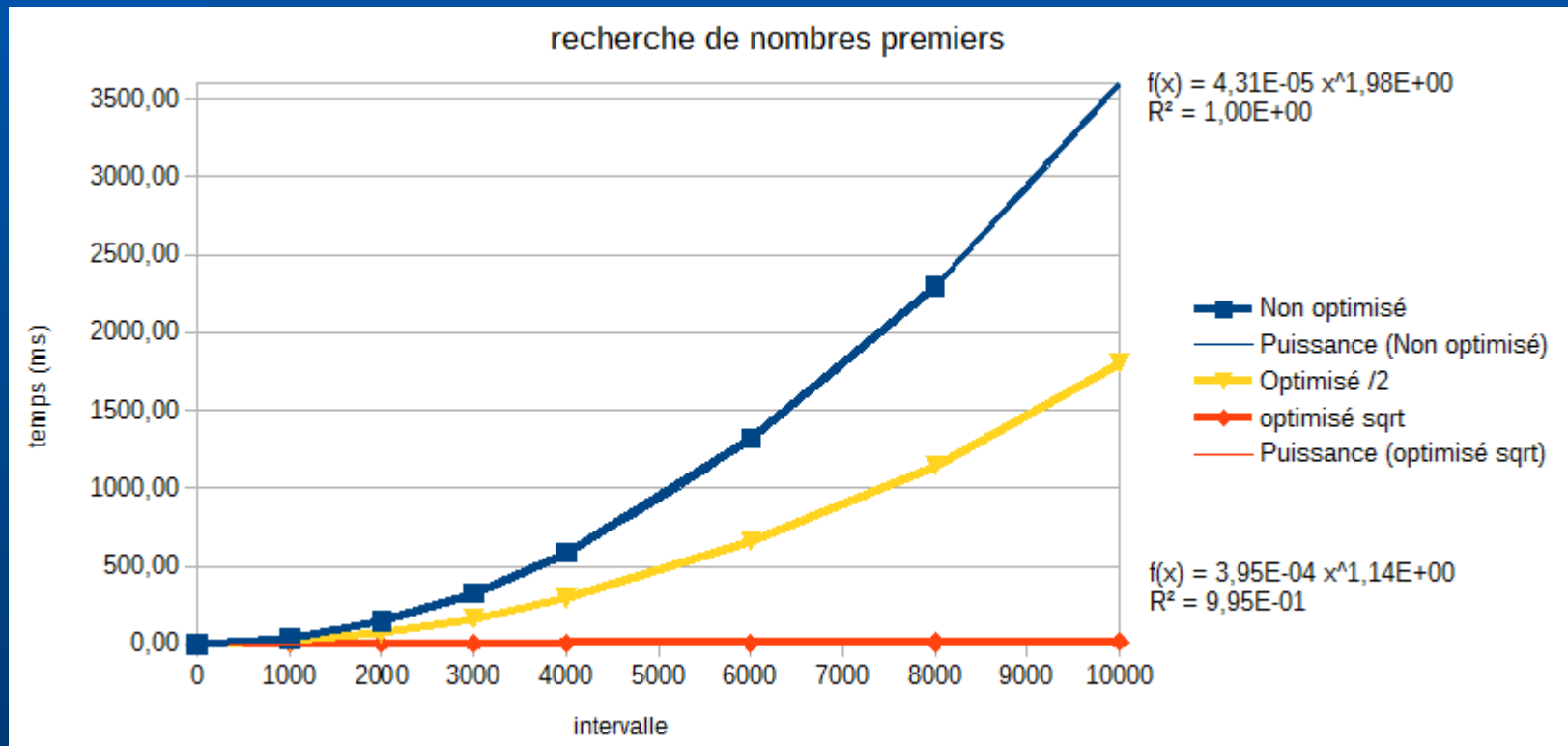
```
    return 0;
}
```

# Algorithmique - bilan

## Informatique et Science du Numérique



### Langage Python



# Algorithmique - bilan

## Informatique et Science du Numérique

