

SINfon

03 XII 2020

While (condition)

{
 =
 =
 =
 =
 } bloc d'instruction
}

TD 2020/11/26

- ① Saisir une durée en [s] (entier)
- ② Décompter les secondes.

Entrée Tc durée : 5 ↪
5
4
3
2
1
0

→ While

printf()
usleep (1000000);

A faire Bon 3. XII. après-midi

• ./app (5)

$t =$

5
4
3
2
1
<u>5</u> \downarrow

./app Ø

while (cond) {

=

}

$t > 0$.

expression

```
while(t>0) {
    printf("%d\n", t);
    t--;
    usleep(1000000);
}
```

block
d'instruction

TANT QUE expression VRAIE

[BLOC INSTRUCTION]

VRAIE si $t \neq 0$
FAUX si $t = 0$

soit -

while (t) { ... } .

↳ toujours VRAIE \rightarrow boucle ∞

Boucle while

Calculer de manière itérative une valeur approchée de π .

$$\hat{\pi} = 4 \left(\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \dots \right)$$

$n : 1, 3, 5, 7 \dots$
Signe: $+ ; - ; + ; - \dots$

const double pi = 3.141592653589793;

Effectuer le calcul de $\hat{\pi}$ jusqu'à obtenir un écart en valeur absolue < error.

error = 1E-6

pi = 3.14 ...

$\hat{\pi} = \emptyset$

error = 1E-6

TANT QUE $|pi - \hat{\pi}| > error$ ALORS

[affiner $\hat{\pi}$ (n , signe)]

afficher $pi \quad \hat{\pi} \quad |pi - \hat{\pi}|$

```

int main(int argc, char const *argv[]) {

    const double pi = 3.141592653589793;
    double p = 0.; // estimate of pi
    double error = 1e-3;
    uint32_t n = 1;
    int32_t s = +1;
    uint32_t maxNumIter = 1000000;

    // 1. ajouter un compteur sur le nombre d'itération
    // 2. modifier la condition pour s'arrêter si on atteint la limite de l'e
    //     ou une limite du nombre d'itération max : maxNumIter
    // jusqu'à 15:45
    while( fabs(pi-p)>=error) {

        p += s * (4. / n);
        n += 2; // 1, 3, 5, 7...
        s = -s; // +1, -1, +1, -1...

        printf("%+.10lf\n", p);
    }
    return 0;
}

```

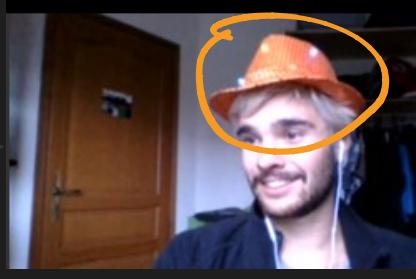
NAL PROBLEMS OUTPUT DEBUG CONSOLE

1: zsh

105845893

125997027

105866176



$$\begin{array}{c|c|c|c}
 a & b & a+b & \bar{a} & \bar{b} \\
 \hline
 0 & 0 & ? & 1 & 1 \\
 0 & 1 & ? & 1 & 0 \\
 1 & 0 & ? & 0 & 1 \\
 1 & 1 & ? & 0 & 0
 \end{array}$$

$\neg(a+b)$ $\neg\bar{a}\&\bar{b}$

Boucle do... While

while:

=====

① on teste une condition

② on exécute un bloc en boucle
selon ①



// Potentiellement le bloc peut ne jamais
être exécuté.

do while:

① on exécute le bloc ↗
② on test la condition ↘

le bloc est exécuté au moins 1x.

do

for

—
—
—
—

for

while (condition);

TD 2020/11 Q3 b)



- demander à l'utilisateur de saisir
une valeur entre < 100

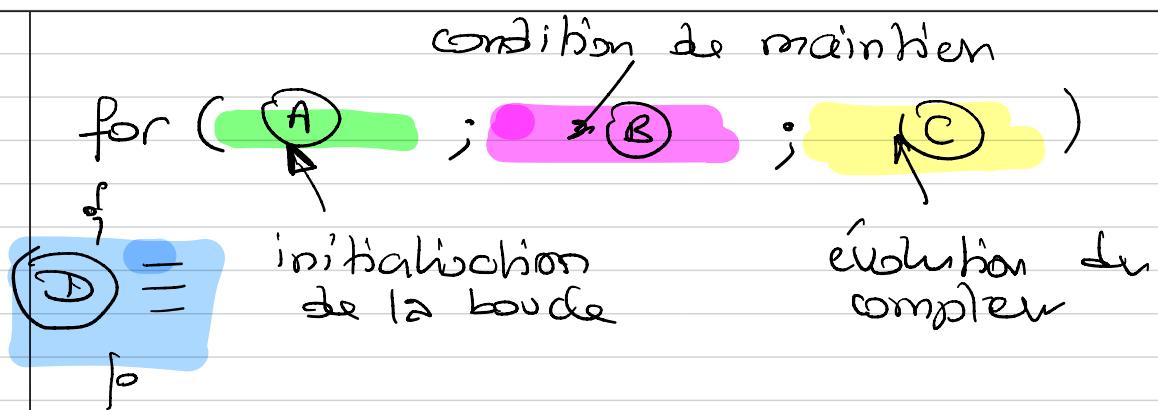
- si ≥ 100 redemander.

En sortie de boucle : afficher la valeur
en caractères.

16'95

```
do {  
    printf("Enter an integer value <100: ");  
    ret = scanf("%d", &n);  
}  
} while (n>=100);
```

Boucle for. → nbre d'itérations connu.



ex: afficher les nombres de 1 à 10

```
for( i = 1 ; i <=10 ; i = i+1 ) {  
    printf( "%d\n", i )  
}
```

Forme: • A

i ← 1

• évaluation de B : i ≤ 10 oui

si B est vrai Alors

(1)

printf(%d)

sinon on sort. (Fin)

• évaluation de C i ← i + 1

• On rentre en B

ici (!) i vaudra 11

faire deux fois: // afficher les nombres pairs entre
 et 100.