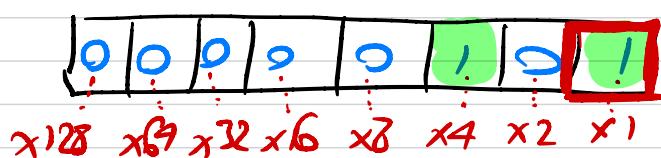


Test si nbre entier pair / impair. 9x1.2020

IMPARI
LSB = 1

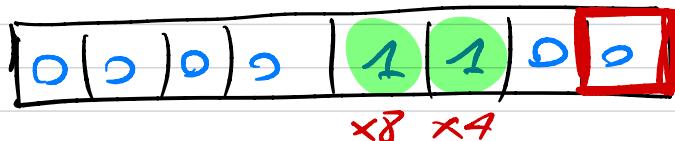
$$i = 5$$



$$4 + 1 = S_{10}$$

PARI

$$i = 12$$



LSB = 0

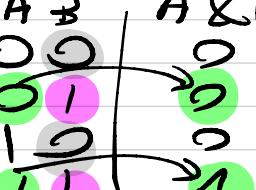
$$8 + 4 = 12_{10}$$

ET LOGIQUE

$i \& \text{mask}$

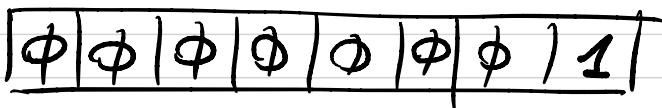
$A \oplus A \& R$

$i :$

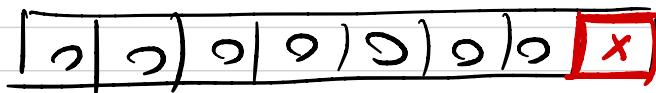


masque

mask:



$i \& \text{mask}$



Si ($i \& \text{mask} == 0$) Alors

i est pair.

Sinon

i est impair.

- Ex:
- Saisir au clavier des caractères
 - Compter ces caractères (sans \n) et retour : \n

> - /app

ABCDEF ↲

6

> /app

12x37 ↲

5

> - /app

aubucudu ↲

8

Boucle.

~~while ... | do... while | for~~

Saisie

Il faut saisir
le caractère
AVANT de le
tester.

on n'
connaît pas
le # d'itérati

uint32_t cpt = 0;
char c = 0;

do {

getchar()

• saisie d'un caractère
dans une variable c

• Incrementer le cpt.

}

while (c != '\n');

cpt--;

Condition de maintien : Tant que la condition est VRAIE on répète dans la boucle.

toto ↲ => cpt = 5

Supprime le \n du compte.

les mystères éclaircis du getchchar.

c = getchchar();

↳ . / app est mis en "pause"

c'est l'OS qui prend la main pour la gestion du clavier.

getchchar() → il attend '\n' avant de poursuivre l'exécution du programme.

1 2 3 \n

c ← '1'

c = getchchar();

c ← '2'

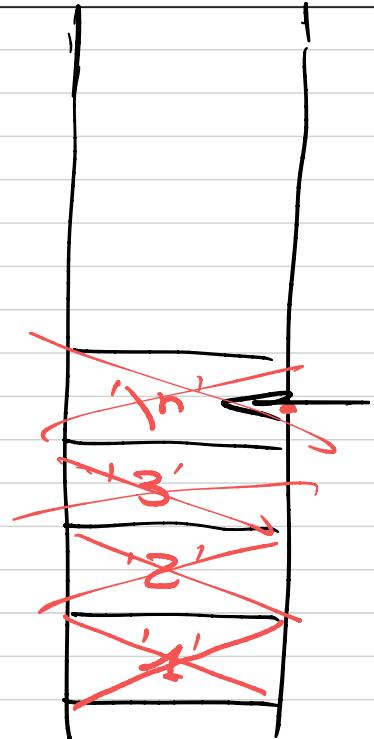
c = getchchar();

c ← '3'

c = getchchar();

c ← '\n'

c = '\n' ⇒ fin de boucle



TD 2020/10/03 b

- Ex: → Savoir un nombre binair
au clavier
- Afficher la valeur correspondante
en base 10 ↗ \rightarrow NON SIGNÉE

• /app
 $\phi\phi11 \leftarrow$

3

• /app
 $\phi \leftarrow$

φ

• /app
 $1111101000 \leftarrow$

1000

14h20

- Savoir : boucle \overline{while} ~~while~~ do..while ~~for~~
- Condition de maintien: $c != '\n'$.
- Conversion des caractères '0' ou '1'
en base 10

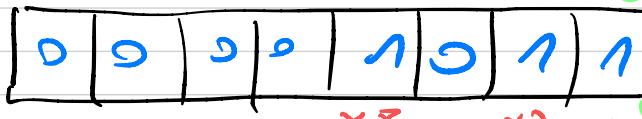
Algorithme



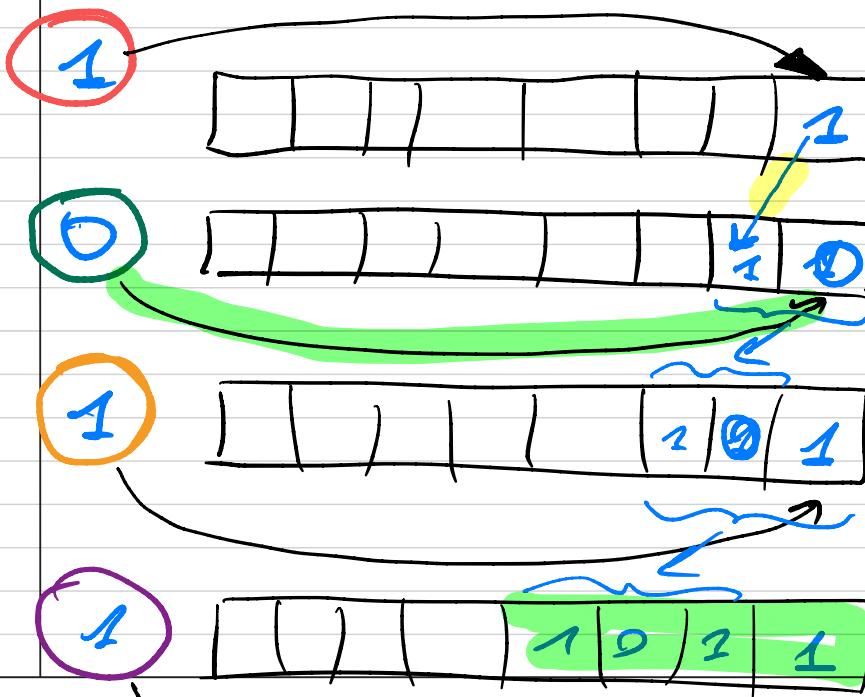
`uint8_t i = 0;`



[boucle saisie +
interprétation]



11_{10}



2 op:

(1) décalage de 1 bit à gauche
(2) inversion du LSR

144
6

Implementation

uint8_t $i = \emptyset$;

char $c = \emptyset$;

do {

$c = \text{getchar}();$

① Vérifier que c vaut ' \emptyset ' ou ' 1 '

② Si c est correct :

2.a décalez d' 1 bit à gauche
le contenu de i

2.b incrémenter 1 ou \emptyset selon la
valeur de c dans le
LSB de i

} while ($c \neq '\backslash n'$);

$\text{printf}(\text{"}\%d\text{u"} , i)$;

{ Terminer l'exercice pour la prochaine
semaine. (16. xi. 2026) .