

✓ Question : Indiquez les commandes pour compiler et exécuter et donnez une trace d'exécution avec votre nom.

```
g++ helloWord.cpp  
./a.out
```

```
Bonjour Rayanne Mellah.  
Oops, je veux dire Mellah.
```

✓ Question : Passer des arguments en ligne de commande à votre programme et décrivez ce qui se passe.

Le code imprime tous les arguments

```
rayanne@rian:~/Documents/BUT2/R305/TDP4/C++$ ./a.out a b c d e f  
0 - ./a.out  
1 - a  
2 - b  
3 - c  
4 - d  
5 - e  
6 - f
```

Téléchargez le programme testUnion.cpp

✓ Question : compilez et expliquez l'exécution.

Ce programme explore l'utilisation d'une union pour stocker et manipuler des données de plusieurs manières dans un même espace mémoire. Il initialise un entier (u.i), puis affiche sa valeur et ses octets individuels. Ensuite, il modifie un octet, affectant la valeur entière, et utilise des champs bit à bit pour manipuler des portions spécifiques de l'entier.

1 Conversion

Reprenez vos cours de BUT 1 expliquant la représentation des réels en mémoire. Vous connaissez la représentation en virgule flottante https://fr.wikipedia.org/wiki/IEEE_754 :



Nous travaillerons avec la version 32 bits (simple précision)

✓ Question : En vous inspirant de testUnion.cpp, écrivez un programme (copier ici le code) qui extrait les informations

- mantisse sur 23 bits
- exposant sur 8 bits
- signe sur 1 bit

```
#include<stdio.h>  
#include <stdlib.h>  
  
union U{  
    unsigned int nb;  
    float nb2;  
};  
  
int  
main(int argc, char **argv, char **envp){  
    if(argc <= 1){  
        fprintf(stdout,"no arguments\n");  
        return 1;  
    }  
    U u;  
    sscanf (argv [1], "%f", &u.nb2);  
    fprintf(stdout, "nb : %X\n",u.nb);  
    unsigned int signe = (u.nb >> 31) & 0x1;  
    unsigned int exposant = (u.nb >> 23) & 0xFF;  
    unsigned int mantisse = u.nb & 0x7FFFFFFF;  
    exposant -= 127;  
    fprintf(stdout, "signe : %X\n",signe);  
    fprintf(stdout, "exposant : %X\n",exposant);  
    fprintf(stdout, "mantisse : %X\n",mantisse);  
  
    fprintf(stdout,"\n");  
    return 0;  
}
```

✓ Question : Donnez les représentations des : signe, mantisse et exposant des cas particuliers :

type	valeur
Zéro	0
Plus petit nombre normalisé	$1,175\,494\,35 \times 10^{-38}$
Un	1
Plus grand nombre normalisé	$3,402\,823\,46 \times 10^{38}$
Infini	?
Not a Number	NaN

Répondre ici ?

2 Mémoire dynamique

Le langage C permet de distinguer différents type de mémoire :

- « statique » limité à un module de compilation (on parlerait de classe en Java)
- « externe » ou variable globale
- « paramètre » toujours passer par valeur dans les fonctions. En réalité en four-10 nissant une référence ou un pointeur il est possible de faire des effets de bords.
- « automatique » ou variable local à une fonction
- « registre » qui n'est une indication (pas très suivi) d'optimisation d'une variable locale

Mais il arrive que l'on ne connaisse pas à l'avance le nombre ou la nature des informations que l'on souhaite manipuler.

On trouve alors des fonctions en langage C : malloc et free. Ils sont remplacés par deux opérateurs en langage C++ : new et delete.

Voici la façon de les utiliser pour unique élément :

```
int *pi;  
pi = new int;  
pi = 3;  
delete pi;
```

5

Dans le cas de tableau on utilisera la variante avec crochets.

```
int *ti;  
ti = new int [2];  
ti[0] = 3; ti[1] = 4  
delete[] ti;
```

Téléchargez le programme *testAdr.cpp*

✓ Question : Indiquez les commandes pour compiler et exécuter et donnez une trace d'exécution avec du programme.

Répondre ici ?

Question bonus.

✓ Question : Écrivez et donnez le code d'un programme qui demande la taille N d'une suite de Fibonacci et crée dynamiquement un tableau de ces N valeurs. Faire une boucle qui demande un nombre inférieur à N et donne la valeur correspondante.

```
n > 1 => F(n) = F(n-1) + F(n-2)  
n < 2 F(n) = 0
```

Répondre ici ?