

0.1 Exercice 1

Type	Format de printf	Taille en octet	Taille en bits
char	%d	1	8
unsigned char	%u	1	8
short	%d	2	16
unsigned short	%u	2	16
int	%d	4	32
unsigned int	%u	4	32
long	%d	10	80
long long	%lld (C99)	10	80
float	%f; %g; %e	4	32
double	%f; %g; %e	10	80

TABLE 1 – Types.csv

On remarque au niveau des adresses qu'elles sont codées en hexadécimal et que les adresses ne sont pas collées entre les variables.

Type	Adresse début	Taille	Adresse fin
char	0x7fff4fb30490	8	0x7fff4fb30498
unsigned char	0x7fff4fb304a0	8	0x7fff4fb304a8
short	0x7fff4fb304b0	16	0x7fff4fb304c0
unsigned short	0x7fff4fb304c0	16	0x7fff4fb304d0
int	0x7fff4fb304d0	32	0x7fff4fb304f0
unsigned int	0x7fff4fb304e0	32	0x7fff4fb30500
long	0x7fff4fb30500	80	0x7fff4fb30550
long long	0x7fff4fb30510	80	0x7fff4fb30560
float	0x7fff4fb304f0	32	0x7fff4fb30510
double	0x7fff4fb30520	80	0x7fff4fb30570

TABLE 2 – Adresses.csv

0.2 Exercice 2

Le fait que les factoriels calculés soient incorrects est dû au fait qu'il y ait un dépassement de capacité. Afin de palier à cela, le `long long unsigned int` a été utilisé. Après modifications, le factoriel le plus élevé calculé correctement est le factoriel 20. Afin d'afficher les factoriels de 0 à 100, il suffit d'exécuter la commande `./exo2 {0..100}` où la partie `{0..100}` est interprétée par le Shell.