Adresses mémoires

I – Variables dans la pile

1. Dans la fonction main, copier le code suivant et observer le résultat :

```
int a = 0;
printf("main - id: a, value:%d, address:%x, size:%d\n", a, &a, sizeof(a));
```

2. Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
int b = 1;
int c = 2;
int d = 3;
printf("a->b address position difference:%d\n", (long)&b - (long)&a);
printf("b->c address position difference:%d\n", (long)&c - (long)&b);
printf("c->d address position difference:%d\n", (long)&d - (long)&c);
```

Que peut-on dire des positions d'allocation de a, b, c et d en RAM ?

3. Au dessus de la fonction main, copier le code suivant :

```
void Func1(int a) {
	printf("Func1 - id: a, value:%d, address:%x, size:%d\n", a, &a, sizeof(a));
}
```

Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
Func1(a);
```

Que peut-on dire sur les positions d'allocation de a dans main et de a dans Func1 ?

II – Références

1. Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
int &e = a;
printf("main - id: e, value:%d, address:%x, size:%d\n", e, &e, sizeof(e));
```

Que peut-on dire sur a et e dans main ?

2. Au dessus de la fonction main, copier le code suivant :

```
void Func2(int &a) {
	printf("Func2 - id: a, value:%d, address:%x, size:%d\n", a, &a, sizeof(a));
}
```

Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
Func2(a);
```

Que peut-on dire sur les positions d'allocation de a dans main et de a dans Func2 ?

III – Tableaux

1. Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
int tab[2] = { 5,10 };
printf("main - id: tab, value:%x, address:%x, size:%d\n", tab, &(tab), sizeof(tab));
```

Que peut-on dire sur la valeur de tab dans main ?

2. Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
printf("main - id: tab[0], value:%d, address:%x, size:%d\n", tab[0], &(tab[0]),
sizeof(tab[0]));
printf("main - id: *(tab), value:%d, address:%x, size:%d\n", *tab, tab,
sizeof(*tab));
```

Que peut on dire sur le résultat de tab[0] et *(tab) dans main ?

3. Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
printf("main - id: tab[1], value:%d, address:%x, size:%d\n", tab[1], &(tab[1]),
sizeof(tab[1]));
printf("main - id: *(tab + 1), value:%d, address:%x, size:%d\n", *(tab + 1), tab + 1,
sizeof(*(tab + 1)));
```

Que peut-on dire l'opérateur « + » sur une adresse mémoire ?

4. Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
int *ptab = tab;
printf("main - id: ptab[0], value:%d, address:%x, size:%d\n", ptab[0], &(ptab[0]),
sizeof(ptab[0]));
```

Que peut on dire sur le pointeur ptab et le tableau tab dans main ?

5. Au dessus de la fonction main, copier le code suivant :

```
void Func3(int *tab) {
	printf("Func3 - id: tab, value:%d, address:%x, size:%d\n", tab[0], &(tab[0]),
sizeof(tab[0]));
}
```

Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
Func3(tab);
```

Que peut on dire sur tab de la fonction main et tab de la fonction Func3 ?

I – Variables dans le tas :

1. Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
	int* tab2 = new int[10];
	printf("main - id: tab2, value:%x, address:%x, size:%d\n", tab2, &tab2,
sizeof(tab2));
	tab2[0] = 15;
	printf("main - id: tab2[0], value:%d, address:%x, size:%d\n", tab2[0], &(tab2[0]),
sizeof(tab2[0]));
```

Que peut on dire sur le pointeur tab2 de la fonction main ?

2. Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
	int *previous = tab2;
	for (int i = 0; i < 10; ++i) {
		tab2 = new int[i];
		printf("previous->tab2 address position difference:%d\n", (long)tab2 -
(long)previous);
		previous = tab2;
	}
```

Que peut on dire sur la position des espaces alloués dynamiquement dans le tas ?

3. Au dessus de la fonction main, copier le code suivant :

```
typedef struct s { int x,y,z,w; } S;
```

Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
	S *ps = new S;
	printf("main - id: ps, value:%x, address:%x, size:%d\n", ps, &ps, sizeof(ps));
	printf("main - id: (*ps), address:%x, size:%d\n", &(*ps), sizeof(*ps));
```

Que peut on dire sur ps de la fonction main ?

4. Au dessus de la fonction main, copier le code suivant :

```
	void Increment(S *ps) {
		(*ps).x++;
	}
```

Dans la fonction main, copier le code suivant à la suite et observer le résultat :

```
	(*ps).x = 20;
	Increment(ps);
	printf("main - id: (*ps).x, value:%d\n", (*ps).x);
```