

TD 1 - Types, Expressions, variables, printf(), scanf()

Dans ces exercices, on suppose être sur une architecture classique à 1, 2, 4 et 8 octets pour les char, short, int et long, et 4 et 8 octets pour les float et double. On rappelle que l'opérateur sizeof retourne le nombre d'octets utilisés par la machine pour représenter un objet.

Exercice 1 (Codage des entiers) Pour chacun des entiers suivants dire quel est le plus petit type entier C qui permet de les coder. Préciser la représentation machine de cette valeur dans ce type.

- | | |
|--------|--|
| • 0 | • -128 |
| • 1 | • 256 |
| • -2 | • 2 147 483 649 (c'est $2^{31} + 1$) |
| • 192 | • - 2 147 483 649 |
| • -192 | • 4 294 967 295 (c'est $2^{32} - 1$) |
| • 127 | • 18 446 744 073 709 551 615 (c'est $2^{64} - 1$) |
| • 128 | • 20 000 000 000 000 000 000 |

Exercice 2 (Expressions littérales et types) Donnez le type et la valeur des expressions littérales C suivantes :

- | | |
|------------------------------------|--|
| • 1048576 (c'est 2^{20}) | • 2147483648 |
| • 1025 (c'est $2^{10} + 1$) | • 2147483648U |
| • 32 | • 'A' |
| • 032 | • '\n' |
| • 0x32 | • 1.5 |
| • 0xA2F | • 18446744073709551616 (c'est 2^{64}) |
| • 1025L | • 18446744073709551615 |
| • 2147483647 (c'est $2^{31} - 1$) | |

Exercice 3 (Types et changements de type implicite) Donner les types et valeurs des expressions C suivantes. Précisez si le calcul de la valeur engendre un dépassement de capacité.

- | | |
|-------------------|------------|
| • 2147483647 + 1 | • 11. / 3 |
| • 2147483647 + 1L | • 11 % 3 |
| • -2147483647 | • 11. % 3 |
| • -2147483648 | • 2 > 5 |
| • -2147483647 - 1 | • 0 == 0 |
| • 2 + 2.3F | • 12 && 23 |
| • 2 + 0. | • -0 0. |
| • 11 / 3 | • ! 'a' |

Exercice 4 (Affectation) Soient les déclarations suivantes :

```

1 int n=5, p=9, q;
2 short sn=1024, sp=32, sq;
3 float x;
4 char c;
```

Dire pour chaque affectation suivante, quel est le type et la valeur de l'expression à droite de l'affectation et quelle sera la valeur de la variable affectée après affectation. Préciser si des conversions implicites et/ou explicites sont mises en œuvre dans ces affectations.

```

1 q = n<p;
2 q = n==p;
3 q = p%n+p>n;
4 sq = sp+sn;
5 sq = sn*sp;
6 x = p/n;
7 x = (float)p/n;
8 x = (p+0.5)/n;
9 x = (int)(p+0.5)/n;
10 c = 'A'+1;
11 c = 800 ;

```

Exercice 5 (Affichage) Donner le résultat des instructions suivantes :

- `printf("\tHello\n");`
- `printf("%d est le code de %c\n",'A','A');`
- `printf("\\"%s\" occupe %ld octets\n","c'est l'été !",sizeof("c'est l'été !"));`

Exercice 6 (Saisie) Soit l'extrait de code :

```

1 int n;
2 float x;
3 char c;
4 scanf("%d%c%f",&n,&c,&x);

```

Donner le résultat de l'exécution de la commande `scanf` et les valeurs des variables `n`, `x` et `c` pour les saisies suivantes :

- 23a2.5 <Entrée>
- 23a <Entrée> 2.5 <Entrée>
- 23 a 2.5 <Entrée>
- 2.5a 23<Entrée>
- .5a 23<Entrée>

Exercice 7 (Lecture de code en C) Pour les programmes suivants, dire si ils sont corrects ou non. Si vous détectez des erreurs, les corriger. Ensuite, donner le résultat affiché.

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int a,b;
6     a= (1+2*3+8)/5+7;
7     b=3*4*5-4;
8     printf("a = %d, b = %d \n ", a, b);
9     return 0;
10 }

```

```

1 #include <stdio.h>
2
3 int main() {
4     int a,b;
5     a=2;
6     b=a++;
7     printf("a= %d, b = %d \n ", a, b);
8     b+=a+2;
9     a+=b--;
10    printf("a= %d, b = %d \n ", a, b);
11    return 0;
12 }

```

```

1 #include <stdio.h>
2
3 int main() {
4     int a,b; float c;
5     a=2; b=3; c=1.5;
6     d=a+b+c;
7     a=d;
8     c+d=a-b;
9     printf("a= %d, b = %d \n ", a, b);
10    printf("c= %d, d = %d \n ", c, d);
11    return 0;
12 }

```

```

1 #include <stdio.h>
2
3 int main() {
4     float a,b,c;
5     a=1.6;
6     b=2*a+0.8;
7     c=b/3;
8     printf("a= %d \n ", a);
9     printf("a= %d \n ", b);
10    printf("a= %d \n ", c);
11    return 0;
12 }

```