

Programmation en C

Corrigé TD1

22 août 2008

```
1
2 /* Mon premier programme en C
3 ----- */
4
5 #include <stdio.h>
6
7 int main ()
8 {
9     printf("Hello World !!\n");
10    return 0;
11 }

1 /* Jouer avec printf
2 ----- */
3
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 int main(){
8
9     char caractere = 'A';
10    int i = 33;
11    short j = 127;
12    int k = 250;
13
14    printf("%c\n",caractere);
15    printf("%d -> %.8x\n", i, i);
```

```

16
17 while (k>=-20) {
18     printf("%d F -> %.2f C\n",k,(5.0)/9*k-160/9);
19     /* remplacer le %.2f par %.6f */
20     k-=10;
21 }
22
23 printf("\n");
24
25 while (j>-127) {
26     printf("Caractère %c -> code ASCII %d\n",j,j);
27     j--;
28 }
29
30 return EXIT_SUCCESS;
31 }

1 /* Représentation des entiers courts signés et non signés
2 ----- */
3
4 #include <stdlib.h>
5 #include <stdio.h>
6
7 int main() {
8
9     short s = 32767;
10    unsigned short us = 32767;
11
12    printf("s=%d (%x) et us=%d (%x)\n",s,s,us,us);
13
14    s++; us++;
15
16    printf("s=%d (%x) et us=%d (%x)\n",s,s,us,us);
17
18    s = 65535; us = 65535;
19
20    printf("s=%d (%x) et us=%d (%x)\n",s,s,us,us);
21

```

```

22  s++; us++;
23
24  printf("s=%d (%x) et us=%d (%x)\n",s,s,us,us);
25
26
27  return EXIT_SUCCESS;
28 }

1  /* Opérations binaires
2  ----- */
3
4  #include <stdio.h>
5  #include <stdlib.h>
6
7  #define w 32
8
9  /* Le paramètre w précise le nombre de bits du mot w=32 par exemple */
10 /* Le paramètre x donne la variable sur laquelle effectuer la rotation */
11 /* Le paramètre y donne le nombre de bits à shifter */
12 #define ROTL(x,y) (((x) << ((y)&(w-1))) | ((x) >> (w-((y)&(w-1)))))
13 #define ROTR(x,y) (((x) >> ((y)&(w-1))) | ((x) << (w-((y)&(w-1)))))
14
15 int main(){
16
17  int i = 64;
18  int isav = i;
19  int j = 234;
20  int jsav = j;
21  unsigned int sum = 0;
22  unsigned int sum1 = 0;
23
24  while (i>0) {
25    printf("%d",i%2);
26    i=i/2;
27  }
28
29  printf("\n%d -> %x \n", isav, isav);
30

```

```

31 printf("Taille d'un entier = %d",8*sizeof(int));
32
33 isav = ROTR(isav,1);
34
35 printf("\n%d -> %x \n", isav, isav);
36
37 isav = ROTL(isav,4);
38
39 printf("\n%d -> %x \n", isav, isav);
40
41 printf("j=%d\n",j);
42
43 while(j>0){
44     sum = sum ^ (j&1);
45     j=j>>1;
46 }
47
48 printf("Le xor des bits de %d (%x) vaut %d \n",jsav,jsav,sum);
49
50 jsav = jsav ^ (jsav>>16);
51 jsav = jsav ^ (jsav>>8);
52 jsav = jsav ^ (jsav>>4);
53 jsav = jsav ^ (jsav>>2);
54 jsav = jsav ^ (jsav>>1);
55
56 printf("Deuxième calcul = %d\n", jsav&1);
57
58 return EXIT_SUCCESS;
59
60 }

1 /* Tri de 3 éléments
2 ----- */
3
4 #include <stdlib.h>
5 #include <stdio.h>
6
7 int main ()

```

```

8 {
9   int A = 20;
10  int B = 13;
11  int C = 17;
12
13  if (A<B) {
14    if (C<A) printf("%d %d %d\n",C,A,B);
15    else
16      if (C<B) printf("%d %d %d\n",A,C,B);
17      else printf("%d %d %d\n",A,B,C);
18  }
19  else { /* n1 >= n2 */
20    if (C<B) printf("%d %d %d\n",C,B,A);
21    else
22      if (C<A) printf("%d %d %d\n",B,C,A);
23      else printf("%d %d %d\n",B,A,C);
24  }
25
26  return EXIT_SUCCESS;
27 }

1 /* Est-ce que l'addition est commutative en C
2 ----- */
3
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 int main(){
8
9   float somme = 0;
10  float somme1 = 0;
11  int maxsav,max;
12  int n=1;
13
14  scanf("%d",&max);
15  maxsav=max;
16
17  while (max>=1) {

```

```

18     somme = somme + 1.0/max;
19     max--;
20 }
21
22 while (n<=maxsav) {
23     somme1 = somme1 + 1.0/n;
24     n++;
25 }
26
27 printf("la première somme vaut %f et la seconde vaut %f\n",somme,somme1);
28
29 return EXIT_SUCCESS;
30 }

```

```

1 /* Programme de calcul de PGCD de deux entiers
2 ----- */
3
4 #include <stdlib.h>
5 #include <stdio.h>
6
7 int main ()
8 {
9     int a,b,r,x,y;
10    x = a = 51;
11    y = b = 12;
12
13    while (b != 0) {
14        r = a % b; a = b; b = r; }
15
16    printf("pgcd (%d,%d) = %d\n",x,y,a);
17
18    return EXIT_SUCCESS;
19 }

```