

**Solution 1 :** Hello World!

Il suffit d'inclure la librairie `stdio.h` et de faire appel à la fonction `printf`.

---

```
1 #include <stdio.h>
2 int main () {
3     printf("Hello World!\n");
4     return 0;
5 }
```

---

**Solution 2 :** Conversion Fahrenheit – Celsius

Voici le code qui affiche la table de conversion. Il faut bien penser à utiliser `5.0/9` pour effectuer une division flottante : la division d'entiers donnerait `0`!

---

```
1 #include <stdio.h>
2 int main(){
3     int k = 250;
4
5     while (k >= -20) {
6         printf("%3d F -> %6.2f C\n", k, (5.0)/9*k-160.0/9);
7         k -= 10;
8     }
9     printf("\n");
10    return 0;
11 }
```

---

Il suffit de remplacer le `.2f` par `.6f` dans le `printf` pour afficher 6 décimales. Le `%3d` au lieu de l'habituel `%d` permet d'afficher l'entier sur 3 cases (quelle que soit sa taille) pour garder les flèches alignées. C'est la même chose pour le `%6.2f` qui affiche un flottant sur 6 cases minimum.

**Solution 3 :** Affichage dans l'ordre croissant

Il y a 6 choix possibles pour l'ordre de 3 éléments, donc il faut explorer toutes les possibilités.

---

```
1 #include <stdio.h>
2 int main () {
3     int A = 17;
4     int B = 23;
5     int C = 11;
6
7     if (A < B) {
8         if (C < A) {
```

```

9     printf("%d %d %d\n",C,A,B);
10    } else if (C < B) {
11        printf("%d %d %d\n",A,C,B);
12    }     else {
13        printf("%d %d %d\n",A,B,C);
14    }
15 } else {
16     if (C < B) {
17         printf("%d %d %d\n",C,B,A);
18     } else if (C < A) {
19         printf("%d %d %d\n",B,C,A);
20     }     else {
21         printf("%d %d %d\n",B,A,C);
22     }
23 }
24 return 0;
25 }

```

---

#### Solution 4 : Affichage d'un triangle

```

1 #include <stdio.h>
2 int main(){
3     int n=8;
4     int nbespace = n-1;
5     int nbetoile = 1;
6     int j;
7
8     while(n >= 1) {
9         j=nbespace;
10        while(j >= 1) {
11            printf(" ");
12            j=j-1;
13        }
14
15        j=nbetoile;
16        while(j >= 1) {
17            printf("*");
18            j=j-1;
19        }
20
21        printf("\n");
22
23        nbespace = nbespace-1;
24        nbetoile = nbetoile+2;
25        n=n-1;
26    }
27    return 0;
28 }

```

---

#### Solution 5 : Calcul avec les Entiers

**5.a]** Voici un programme simple qui fait une recherche exhaustive sur tous les triplets. Il est bien sur possible de l'améliorer en utilisant un test plus efficace pour savoir si  $a^2 + b^2$  est un

carré (ici on essaye tous les  $c^2$ ).

---

```
1 #include <stdio.h>
2 #define N 10
3 int main(){
4     int a=1, b=1, c;
5     float d;
6
7     while(a <= N) {
8         b = 1;
9         while(b < a) {
10            d = a*a + b*b;
11            c = 1;
12            while(c*c < d) {
13                c++;
14            }
15            if (c*c == d) {
16                printf("%d^2 + %d^2 = %d^2\n", a, b, c);
17            }
18            b++;
19        }
20        a++;
21    }
22    return 0;
23 }
```

---

**5.b]** Et le calcul itératif du PGCD.

---

```
1 #include <stdio.h>
2 int main () {
3     int a,b,r,x,y;
4     x = a = 51;
5     y = b = 12;
6
7     while (b != 0) {
8         r = a % b;
9         a = b;
10        b = r;
11    }
12    printf("pgcd (%d,%d) = %d\n", x, y, a);
13    return 0;
14 }
```

---

Notez au passage l'instruction `x = a = 51;` qui montre un aspect intéressant du C : l'opération d'affectation retourne la valeur qui vient d'être affectée. Quand le programme exécute cette ligne, il voit le premier `=` qui correspond à une affectation dans `x` et la valeur à affecter est l'évaluation de `a = 51`. L'instruction `a = 51` va elle affecter 51 à `a` et ensuite retourner la valeur affectée, donc 51, qui sera aussi affectée à `x`.