

```
1 #include <stdio.h>
2
3 int main()
4 {
5
6     int A;
7     int c = 0;
8     /*déclaration de la variable "A"
9     qui va permettre plustard de stocker
10    l'année entrée par l'utilisateur*/
11
12     scanf("%d", &A);
13     /* la fonction scanf va permettre
14     l'affectation après execution de la variable A en stockant
15     dedans l'année entrée par l'utilisateur*/
16
17     /*on traite tout d'abord
18     le cas de A non divisible par 4 qui donne
19     bien sur A non bisextile*/
20     if (A % 4 != 0)
21     {
22         printf("%d n'est pas bisextile\n", A);
23     }
24
25     /* Ensuite le else if balaye tous les cas ou
26     A non divisible par 4, reste maintenant les deux cas
27     avec A divisible par 4:
28     1.A divisible par 100 ET pas par 400
29     2.A non divisible par 100 OU divisible par 400
30
31     le cas 2 est juste le contraire logique du cas 1.
32     Le premier else if traite le cas 1 et le dernier else
33     traite le cas 2 qui est le reste de tous les conditions posées avant. */
34     else if (A % 100 == 0 && A % 400 != 0)
35     {
36         printf("%d n'est pas bisextile\n", A);
37     }
38
39     else
40     {
41         printf("%d est bisextile\n", A);
42     }
43 }
44
45
```

```
1 #include <stdio.h>
2 #include <math.h>
3 #define M_PI 3.14159265358979323846
4
5 int main()
6 {
7
8     float x, y, theta, r;
9
10    scanf("%f %f", &x, &y);
11
12    r = sqrt(x * x + y * y);
13
14    if (x > 0)
15    {
16        theta = atan(y / x);
17    }
18    else if (x < 0)
19    {
20        theta = atan(y / x) + M_PI;
21    }
22    else
23    {
24        if (y > 0)
25        {
26            theta = M_PI / 2;
27        }
28        else if (y < 0)
29        {
30            theta = -M_PI / 2;
31        }
32        else
33        {
34            theta = 0;
35        }
36    }
37
38    printf(" les coordonnes polaires sont: r = %f, et theta = %f\n", r, theta);
39    //pour avoir le résultat en degrés écrire 180 * theta / M_PI à la place de theta
40    dans printf
41 }
42
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5
6     int i, n, a, b, c;
7     a = 1;
8     b = 1;
9     i = 2;
10
11     scanf("%d", &n);
12     while (n < 2)
13     {
14         scanf("%d", &n);
15     }
16
17     while (i <= n)
18     {
19         c = a + b;
20         a = b;
21         b = c;
22         i++;
23     }
24     printf("%d \n", c);
25 }
26
27
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5
6     int a, b, c, n;
7     a = 1;
8     b = 1;
9     int i = 2;
10
11     scanf("%d", &n);
12     while (n < 2)
13     {
14         scanf("%d", &n);
15     }
16
17     while (i <= n)
18     {
19         c = a + b;
20         printf("U%d = %d\n", i, c);
21         a = b;
22         b = c;
23         i++;
24     }
25 }
26
27
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int A, i, a, b, c;
6     a = 1;
7     b = 1;
8     i = 1;
9     c = 2;
10
11     scanf("%d", &A);
12
13     while (A >= c)
14     {
15         c = a + b;
16         a = b;
17         b = c;
18         i++;
19     }
20     printf("%d \n", i);
21 }
22
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n;
6     int Spair = 0;
7     int Simpair = 0;
8
9     scanf("%d", &n);
10
11     while (n != 0)
12     {
13
14         if (n % 2 == 0)
15         {
16             Spair = Spair + n;
17         }
18         else
19         {
20             Simpair = Simpair + n;
21         }
22
23         scanf("%d", &n);
24     }
25     printf("La somme paire est de : %d et la somme impaire est de :%d\n", Spair,
26     Simpair);
27 }
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a, b;
6     scanf("%d %d", &a, &b);
7     int q = 0;
8     int r = 0;
9     int s = b;
10
11     while (s <= a)
12     {
13         s = s + b;
14         q++;
15     }
16     s = s - b;
17     r = a - s;
18
19     printf("Le quotient est :%d le reste est : %d\n", q, r);
20 }
21
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5
6     int a;
7     int i = 1;
8     int s = 0;
9
10    scanf("%d", &a);
11
12    while (i < a)
13    {
14        if (a % i == 0)
15        {
16            s = s + i;
17        }
18        i++;
19    }
20
21    if (s == a)
22    {
23        printf("%d est un nombre parfait\n", a);
24    }
25
26    else
27    {
28        printf("%d n'est pas un nombre parfait\n", a);
29    }
30 }
31
```



```
1 #include <stdio.h>
2
3 int main()
4 {
5
6     int a;
7     int i = 1;
8     int s = 0;
9     int n;
10    a = 3;
11    int cpt = 0;
12
13    //scanf("%d", &a);
14    scanf("%d", &n);
15
16    while (cpt < n)
17    {
18        i = 1;
19        s = 0;
20        while (i < a)
21        {
22            if (a % i == 0)
23            {
24                s = s + i;
25            }
26            i++;
27        }
28
29        if (s == a)
30        {
31            printf("%d est un nombre parfait\n", a);
32            cpt++;
33        }
34
35        a++;
36    }
37 }
38
```

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     int n;
7     scanf("%d", &n);
8     int i;
9
10    while (n < 0) //Cet boucle while garantit d'avoir un n positif dans notre
    traitement
11    {
12        scanf("%d", &n);
13    }
14
15    if (n == 1)
16    {
17        printf("%d n'est pas premier\n", n);
18    }
19
20    else
21    {
22        for (i = 2; i <= sqrt(n); i++)
23        {
24            if (n % i == 0)
25            {
26                printf("%d n'est pas premier\n", n);
27                break;
28            }
29        }
30
31        if (i > sqrt(n))
32        {
33            printf("%d est premier\n", n);
34        }
35    }
36 }
37
```

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     int i, m, n;
7     scanf("%d", &m);
8     n = 2;
9     int cpt = 0;
10    while (cpt < m)
11    {
12        for (i = 2; i < sqrt(n); i++)
13        {
14            if (n % i == 0)
15            {
16                break;
17            }
18        }
19
20        if (i > sqrt(n))
21        {
22            printf("%d \n", n);
23            cpt++;
24        }
25
26        n++;
27    }
28 }
29
```

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     int i, m, n;
7     scanf("%d", &m);
8     n = 2;
9
10    while (n < m)
11    {
12        for (i = 2; i < sqrt(n); i++)
13        {
14            if (n % i == 0)
15            {
16                break;
17            }
18        }
19
20        if (i > sqrt(n))
21        {
22            printf("%d \n", n);
23        }
24
25        n++;
26    }
27 }
28
29
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n;
6     scanf("%d", &n);
7     int cpt = 1;
8     int m = n;
9     /*j'ai ajouté la variable "m" pour sauver la valeur de la variable "n" pour
10 l'affichage finale seulement,
11 car la valeur de n va etre modifiée lors du traitement dans la boucle while */
12     while (n > 10)
13     {
14         n = n / 10;
15         cpt++;
16     }
17     printf("le nombre de chiffre de %d est: %d\n", m, cpt);
18 }
19
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n;
6     scanf("%d", &n);
7     int m = n;
8     int s = 0;
9
10    while (n > 0)
11    {
12        s = s + n % 10;
13        n = n / 10;
14    }
15    printf("la somme des chiffres de %d est: %d\n", m, s);
16 }
17
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n;
6     scanf("%d", &n);
7     int m = n;
8     int s = 0;
9
10    while (n > 0)
11    {
12        s = s * 10 + n % 10;
13        n = n / 10;
14    }
15    printf("le nombre %d inverse est: %d\n", m, s);
16 }
17
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int n;
6     scanf("%d", &n);
7     int m = n;
8     int s = 0;
9
10    while (n >= 10)
11    {
12        s = 0;
13        while (n > 0)
14        {
15            s = s + n % 10;
16            n = n / 10;
17        }
18        n = s;
19    }
20    printf("la racine digitale de %d est: %d\n", m, n);
21 }
22
```



```
1 #include <stdio.h>
2 #include <math.h>
3 #include <stdlib.h>
4
5 int main()
6 {
7     float A;
8
9     scanf("%f", &A);
10
11     float x = A;
12
13     while (abs(x - sqrt(A)) > 1e-3)
14     {
15         x = 0.5 * (x + A / x);
16         printf("%f\n", x - sqrt(A));
17     }
18
19     printf("l'approximation de la racine de %f est: %f\n", A, x);
20 }
21
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5
6     int n;
7
8     scanf("%d", &n);
9
10    int u = n;
11    int cpt = 0;
12
13    while (cpt < 2)
14    {
15
16        if (u % 2 == 0)
17        {
18            u = u / 2;
19        }
20        else
21        {
22            u = 3 * u + 1;
23        }
24
25        printf("%d\n", u);
26
27        if (u == 1)
28        {
29            cpt++;
30        }
31    }
32 }
```

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     int n;
7     int i = 2;
8
9     scanf("%d", &n);
10    int m = n;
11
12    while (n < 2)
13    {
14        scanf("%d", &n); // c'est pour garantir d'avoir un n > ou égale à 2.
15    }
16
17    while (i <= sqrt(n)) //on parcourt tous les nombres < ou égale à la racine de n.
18    {
19        int k;
20        /* la boucle for ci-dessous permet de selectionner
21           les nombres premiers < à la racine de n*/
22
23        for (k = 2; k < sqrt(i); k++)
24        {
25
26            if (i % k == 0)
27            {
28                break;
29            }
30        }
31
32        if (k > sqrt(i) && n % i == 0)
33        {
34
35            //si on entre dans ce if sa veut dire que i est premier et divise n,
36            //on cherche ensuite combien de i il y'a dans n, ainsi:
37
38            int cpt = 0;
39            while (n % i == 0)
40            {
41                n = n / i;
42                cpt++;
43                // c'est la variable cpt qui va compter combien de fois on a de i
44                // et c'est ce qu'on veut trouver la puissance ai d'un certain
45                // dans n.
46                // facteur premier pi.
47                // dans notre cas i c'est le pi et ai c'est cpt.
48            }
49            printf("%d %d\n", i, cpt);
50        }
51        i++;
52    }
53    printf("%d 1\n", n);
54 }
55
```