

Solution des Travaux Dirigés 2

Solution d'exercice 1

- **Question 1** : Non il n'est pas lisible
- **Question 2** :

Algorithme 1 Lisible

VAR: A,B,S,I : entier ;

DÉBUT

Lire (A,B);

$S \leftarrow S$;

$I \leftarrow 1$;

tantque $I \leq B$ **Faire**

$S \leftarrow S + I$;

$I \leftarrow I + 1$;

fin tantque

Ecrire (S);

Fin

- **Question 3** : oui elle est judicieuse puisque pour boucle pour il faut savoir le nombre d'itération pour pouvoir l'utiliser. Ce qui concerne la boucle répéter, on fait au moins une itération or rien n'oblige l'utilisateur de donner une valeur de $B < 1$.

Solution d'exercice 2

- **Question 1** : Le déroulement de cet algorithme

Algorithme 2 SommeNbr

VAR: N,nbr,S,I : entier ;

```

1: DÉBUT
2: Ecrire ("donner le nombre d elements");
3: Lire (N);
4: Pour I  $\leftarrow$  1 à N Faire
5:   Ecrire ("donner vos éléments");;
6:   Lire (nbr);
7: fin Pour
8: Pour I  $\leftarrow$  1 à N Faire
9:   S  $\leftarrow$  0;
10:  S  $\leftarrow$  S+nbr ;
11:  Ecrire ("La somme de vos elements est :",S);
12: fin Pour
13: Fin

```

Instruction	I	S	nbr	N	Ecrans
2					donner le nombre d'elements
3				4	
4	1				
5					donner vos éléments
6			10		
4	2				
5					donner vos éléments
6			15		
4	3				
5					donner vos éléments
6			22		
4	4				
5					donner vos éléments
6			8		
4	5				
8	1				
9		0			
10		8			
11					La somme de vos elements est 8
8	2				
9		0			
10		8			

11					La somme de vos elements est 8
8	3				
9		0			
10		8			
11					La somme de vos elements est 8
8	4				
9		0			
10		8			
11					La somme de vos elements est 8
8	4				
13					

- **Question 2 :** Les problèmes de cette solution sont comme suit :
 1. L'utilisation de deux (02) boucle pour est inutile, c'est erreur fatal de cet algorithme
 2. Une variable stock une seule information, donc le fait de mettre une boucle pour (ligne 4-7) ne permet pas de stocker toutes les valeurs (10,15,22,8) mais juste la dernière 8.
 3. L'initialisation de la variable S doit être fait à l'extérieur de la boucle pour

Solution d'exercice 3

Algorithme 3 Exercice3

```

VAR: N,S,i : entier ;
1: DÉBUT
2: Ecrire ("Donner le nombre N");
3: Lire (N);
4: S ← 0 ;
5: Pour i ← 1 à N Faire
6:   S ← S+i;
7:   Ecrire (i,"+");
8: fin Pour
9: Ecrire ("=",S);
10: Fin

```

```

1 #include <stdio.h>
2 main()

```

```

3 {
4     int i,N,S=0;
5     printf("Donner le nombre N\n");
6     scanf("%d",&N);
7     for(i=1;i<=N;i++){
8         S+=i;
9         printf("%d",i);
10        if(i<N)
11            printf("+");
12    }
13    printf("=%d",S);
14 }

```

Donner le nombre N

4

1+2+3+4=10

Solution d'exercice 4

Algorithme 4 Exercice4

VAR: P,N,a,i : entier ;

```

1: DÉBUT
2: Ecrire ("Donner le nombre N");
3: Lire (N);
4: Ecrire ("Donner le nombre a");
5: Lire (a);
6: P ← 1;
7: Pour i ← 1 à a Faire
8:     P ← P*N;
9: fin Pour
10: Ecrire ("P=",P);
11: Fin

```

```

1 #include<stdio.h>
2 main()
3 {
4     int i,N,a,P=1;
5     printf("Donner le nombre N\n");
6     scanf("%d",&N);
7     printf("Donner le nombre a\n");
8     scanf("%d",&a);
9     for(i=1;i<=N;i++){
10        P*=P;

```

```

11 }
12 printf("P=%d",P);
13 }

```

```

Donner le nombre N
4
Donner le nombre a
2
P=16

```

Solution d'exercice 5

Algorithme 5 Exercice5

VAR: M,A,B : entier ;

- 1: **DÉBUT**
- 2: **Ecrire** ("Donner le nombre A");
- 3: **Lire** (A);
- 4: **Ecrire** ("Donner le nombre B");
- 5: **Lire** (B);
- 6: $M \leftarrow A$;
- 7: **tantque** $M \leq B$ **Faire**
- 8: $M \leftarrow M-B$;
- 9: **fin tantque**
- 10: **Ecrire** ("Le reste de la division =",M);
- 11: **Fin**

```

1 #include<stdio.h>
2 main()
3 {
4     int M,A,B;
5     printf("Donner le nombre A\n");
6     scanf("%d",&A);
7     printf("Donner le nombre B\n");
8     scanf("%d",&B);
9     M=A;
10    while (M<=B)
11        M=B;
12    }
13    printf("Le reste de la division =%d",M);
14 }

```

```

Donner le nombre A
4
Donner le nombre B
2
Le reste de la division = 0

```

Solution d'exercice 6

Algorithme 6 Exercice6

```

VAR: M,A : entier ;
1: DÉBUT
2: Ecrire ("Donner le nombre A");
3: Lire (A);
4:  $M \leftarrow A$ ;
5: tantque  $M \leq 2$  Faire
6:    $M \leftarrow M-2$ ;
7: fin tantque
8: Si  $M=0$  alors
9:   Ecrire (A,"est divisible sur 2");
10: Sinon
11:   Ecrire (A,"n'est divisible sur 2");
12: fin Si
13: Fin

```

```

1 #include <stdio.h>
2 main()
3 {
4     int M,A;
5     printf("Donner le nombre A\n");
6     scanf("%d",&A);
7
8     M=A;
9     while (M<=2)
10    M-=2;
11    }
12    if (M==0)
13        printf("%d est divisible sur 2",A);
14    else
15        printf("%d n'est divisible sur 2",A);
16 }

```

```
Donner le nombre A
5
5 n'est divisible sur 2
```

Solution d'exercice 7

Algorithme 7 Exercice7

```
VAR: Min,N,n,i : entier ;
1: DÉBUT
2: Ecrire ("Donner le nombre n");
3: Lire (n);
4: Pour i ← 1 à n Faire
5:   Ecrire ("Donner le nombre ",i," eme nombre");
6:   Lire (N);
7:   Si Min<N ou i=1 alors
8:     Min ← N
9:   fin Si
10: fin Pour
11: Ecrire ("Le minimum est",Min);
12: Fin
```

```
1 #include<stdio.h>
2 main()
3 {
4     int Min,n,N,i;
5     printf("Donner le nombre n\n");
6     scanf("%d",&n);
7     for(i=1;i<=n;i++){
8         printf("Donner le nombre %deme nombre\n",i);
9         scanf("%d",&N);
10        if(Min>N|| i==1)
11            Min=N;
12    }
13    printf("Le minimum est %d",Min);
14
15 }
```

```
Donner le nombre n
4
Donner le nombre 1eme nombre
2
Donner le nombre 2eme nombre
```

```

4
Donner le nombre 3eme nombre
-3
Donner le nombre 4eme nombre
1
Le minimum est -3

```

Solution d'exercice 8

Algorithme 8 Exercice8

```

VAR: N,nbr_div,i : entier ;
1: DÉBUT
2: Ecrire ("Donner le nombre N");
3: Lire (N);
4: nbr_div ← 0 ;
5: Pour i ← 1 à N Faire
6:   Si N mod i = 0 alors
7:     nbr_div ← nbr_div+1 ;
8:   fin Si
9: fin Pour
10: Si nbr_div=2 alors
11:   Ecrire (N,"est un nombre premier");
12: Sinon
13:   Ecrire (A,"n'est un nombre premier");
14: fin Si
15: Fin

```

```

1 #include<stdio.h>
2 main()
3 {
4     int N,nbr_div,i;
5     printf("Donner le nombre N\n");
6     scanf("%d",&N);
7
8     nbr_div=0;
9     for(i=1;i<=N;i++)
10    if(N%i==0)
11    nbr_div++;
12
13    if(nbr_div==2)
14    printf("%d est nombre premier",N);
15    else

```



```

16 printf("%d n'est un nombre premier",N);
17 }

```

```

Donner le nombre N
5
5 est nombre premier

```

Solution d'exercice 9

Algorithme 9 Exercice9

```

VAR: A,B,i : entier;
1: DÉBUT
2: Ecrire ("Donner le nombre A");
3: Lire (N);
4: Ecrire ("Donner le nombre B");
5: Lire (N);
6:  $i \leftarrow A \bmod B$ ;
7: tantque  $i \neq 0$  Faire
8:    $i \leftarrow B$ ;
9:    $B \leftarrow \text{abs}(A - B)$ ;
10:   $A \leftarrow i$ ;
11: fin tantque
12: Ecrire ("PGCD = ",B);
13: Fin

```

```

1 #include <stdio.h>
2 #include <math.h>
3 main()
4 {
5     int A,B,i;
6     printf("Donner le nombre A\n");
7     scanf("%d",&A);
8     printf("Donner le nombre B\n");
9     scanf("%d",&B);
10    i=A%B;
11    while(i!=0){
12        i=B;
13        B=abs(A-B);
14        A=i;
15        i=A%B;
16    }
17    printf("le PGCD = %d",B);
18 }

```

```
Donner le nombre A
5
Donner le nombre B
15
le PGCD = 5
```

Solution d'exercice 10

```
1 #include<stdio.h>
2 #include<math.h>
3 main()
4 {
5     int n,f,i,j;
6     float S;
7     printf("Donner le nombre n\n");
8     scanf("%d",&n);
9     S=1;
10    printf("\nS_n = 1");
11    for(i=1;i<=n;i++){
12        f=1;
13        for(j=1;j<=i;j++)
14            f*=j;
15        if(i%2==0)
16            printf("+");
17        printf("%d/%d", (int)pow(-1,i), ((2*i+1)*f));
18        S+=pow(-1,i)/((float)(2*i+1)*f);
19    }
20    printf("=%f",S);
21 }
```

```
Donner le nombre n
4
S_n = 1-1/3+1/10-1/42+1/216=0.747487
```

Solution d'exercice 11

```
1 #include<stdio.h>
2 main()
3 {
4     int n=0,a,b,c,d,tmp,i;
5     do
6     {
7         printf("Donner un nombre a 4 chiffres\n");
8         scanf("%d",&n);
9     } while(n<1000);
```

```
10  a=n/1000;
11  b=n%1000/100;
12  c=n%1000%100/10;
13  d=n%1000%100%10;
14  tmp=a;
15  a=d;
16  d=tmp;
17  tmp=b;
18  b=c;
19  c=tmp;
20  printf("%d%d%d%d",a,b,c,d);
21
22 }
```

Donner un nombre a 4 chiffres

2345

2 3 4 5