**Analyse question TP**

**Question 3**

Analyse algorithme : 1+1/2+1/3+1/4

|  |  |  |
| --- | --- | --- |
| Somme 🡨0  I🡨1  Somme 🡨somme+1/1  I🡨I+1  Somme🡨somme+1/2  I🡨I+1  Somme🡨somme+1/3  I🡨I+1  Somme🡨somme+1/4 | Rechercher les éléments qui varient et mettre une variable « i »  Identifier les actions initiales Somme 🡨0  I🡨1  Identifier les actions itératives Somme 🡨somme+1/1  I🡨I+1  Déterminer la condition de fin  Somme 🡨0  I🡨1  Tantque (i<=4)  Somme🡨somme+1/I  I🡨I+1  fintantque | var somme=0,i=1;  while (i<=4){      somme=somme+1/i      i=i+1  }  console.log("la somme est: "+somme) |

**Question 4**

Analyse algorithme : 1/1+(1+2) /(1\*2)+(1+2+3) /(1\*2\*3)+

|  |  |  |
| --- | --- | --- |
| Total 🡨0  Somme🡨1  Produit🡨1  Total🡨total+somme/produit (0+1/1)  Somme🡨somme+2 (1+2)  Produit🡨produit\*2 (1\*2)  Total🡨total+somme/produit 1/1+(1+2)/(1\*2)  Somme🡨somme+3  Produit🡨produit\*3 | Total 🡨0  Somme🡨1  Produit🡨1  I🡨2  Total🡨total+somme/produit  I🡨I+1  Somme🡨somme+I  Produit🡨produit\*I  Total🡨total+somme/produit  I🡨I+1  Somme🡨somme+I  Produit🡨produit\*I | var total=0  var somme=1  var produit=1  var i=2  while (i<=3) {    total=total+somme/produit    somme=somme+i    produit=produit\*i    i=i+1  } |

1/1+(1+2) /(1\*2)+(1+2+3) /(1\*2\*3)+…

On raisonne comme suit :

* le numérateur c’est une somme qui évolue : SOMME (1+2)
* le dénominateur c’est un produit qui évolue : PRODUIT (1\*2)
* il faut une variable pour le total général : TOTAL

**Question 5**

Nombre premier : nombre divisible par 1 et par lui-même uniquement.

3 parties :

1. Trouver si un nombre est premier
2. En trouver 100
3. Stocker dans un tableau et les afficher

Analyse :

Compteur=0 //compteur de nombre premier

Nombre=2 //le premier nombre attesté (1er nombre premier)

Listepremiers=new Array()

Tantque (compteur<=100)

**Chercher si nombre est premier**

var premier=true

var nombre=22

var i=2

while (i<nombre){

    if (nombre%i==0) premier=false

    i=i+1

}

if (premier) {

    LISTEPREMIERS[compteur]=nombre ;

Compteur++

}

**Si oui on l’ajoute à un array et on incrémente le compteur**

    LISTEPREMIERS[compteur]=nombre ;

**Nombre++**

Compteur++

Fintantque

Chercher si nombre est premier

|  |  |  |  |
| --- | --- | --- | --- |
| Premier=true  Si 5%2==0 premier=false  Si 5%3==0 premier=false  Si 5%4==0 premier=false  Premier (true) | Premier=true  Nombre=5  Si nombre%i==0 premier=false  I🡨i+1  Si nombre%i==0 premier=false  I🡨i+1  Premier (true) | Premier=true  Nombre=5  I🡨2  Tant que (i<nombre)  Si nombre%i==0 premier=false  I🡨i+1  Fintant que  Si (premier) afficher (premier) sinon | var premier=true  var nombre=5  var i=2  while (i<nombre){      if (nombre%i==0) premier=false      i=i+1  }  if (premier) {      console.log (nombre+" est premier")  } else { console.log("pas premier")} |

Premier=true

Si 6%2==0 premier=false

Si 6%3==0 premier=false

Si 6%4==0 premier=false

Premier (false) pas un nombre premier

**Question 6**

* Déterminer le nombre de caractère (hors point et espace).

« j e v a i s a u m a r c h e . »

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

On va lire les caractères de la phrase 1 par 1 et vérifier s’ils sont différents de l’espace et du point.

|  |  |
| --- | --- |
| Tantque (carLue! « .» ) //(caractère lue)  Si (carLue != « .» et carLue != «  ») compteur++  fintantque | var phrase ="je vais au marche."  var i=0 //indice du caractère  var compteur=0 //compteur nbr caractère  while (phrase[i]!=".") {      console.log(phrase[i])      if (phrase[i]!="." && phrase[i]!=" ") compteur++      i++  }  console.log("nbr car: "+compteur) |

* Calculer le nombre de mots dans la phrase.

|  |  |
| --- | --- |
| Phrase=« Je vais au marché. »  Compteur=0  I=0  Tanque (carLu != ‘’.’’)  //lecture des espaces  Tantque (carLu==’’ ‘’) je lis le caractère suivant  Tant que (carLu !=’’ ‘’) je lis le caractère suivant  Compteur++  Fintantque | var phrase ="je vais au marche ce jeudi."  var compteur=0  var i=0  while (phrase[i]!=".") {      while (phrase[i]==" " && phrase[i]!=".") i++      while (phrase[i]!=" " && phrase[i]!=".") i++      compteur++  }  console.log("nombre caractère: "+compteur) |

Un mot commence après qu’il n’y ait plus d’espace et s’achève lorsqu’on rencontre un espace.

* Déterminer le nombre d’occurrence d’une lettre dans la phrase. 🡪 PAS FAIT

**Question 7**

* **Réaliser un programme qui permet de déterminer si le mot s’écrit de la même façon à l’envers et a à l’endroit**

|  |  |  |
| --- | --- | --- |
| Var mot=”radar”  Var longueur=5  Var palyndrome=true  Si mot[0] !=[longueur-1] palyndrome=false  Si mot[1] !=[longueur-2] palyndrome=false  Si mot[2] !=[longueur-3] palyndrome=false  Si palyndrome=true afficher(ok) sinon afficher (pas ok) | Var mot=”radar”  Var longueur=5  Var palyndrome=true  i=0  k=longueur-1  tantque (i<k)  Si mot[i] !=[k-1] palyndrome=false  I=i+1  K=k-1  Si palyndrome=true afficher(ok) sinon afficher (pas | var mot="bonjour"  var longueur=7  var palyndrome=true  var i=0  var k=longueur-1  while (i<k) {      if (mot[i]!=mot[k]) palyndrome=false      i=i+1      k=k-1  }  if (palyndrome) console.log("ok"); else console.log("pas ok") |

**Question bonus**

Créer un tableau de la classe et le trier par âge

Tab=[‘’Agbebou’’,30],[‘’Luc’’,21],[‘’Elodie’’,33],[‘’paul’’,44]

var tab=[‘’Agbebou’’,30],[‘’Luc’’,21],[‘’Elodie’’,33],[‘’paul’’,44]

var i=0

while(

function tritableau(tab){

var i=0;

//variable booleenne pour controler si interversion

var interv=true

while (interv) {

    //on se positionne en debut de tableau

    var i=0

    interv=false

    //parcours du tableau

    while (i<tab.length) {

        if (tab[i]>tab[i+1]){

            resultat=tab[i+1];

                interv=true

            tab[i+1]=tab[i];

            tab[i]=resultat;

        }

        i=i+1

    }

}

return tab

}

console.log(tritableau(tab))

**Question 8**

* **Réaliser un programme qui va déterminer le maximum et le minimum dans cet array**

|  |  |  |
| --- | --- | --- |
| TAB | I | MAX |
| 15 | 0 | 15 |
| 17 | 1 |  |
| 1 | 2 |  |
| 3 | 3 |  |
|  | | |
| **max🡨tab[0]**  Si max<tab[1] max🡨tab[1]  Si max<tab[2] max🡨tab[2]  Si max<tab[3] max🡨tab[3] | I<0  Max🡨tab[i]  Tantque(i<4) {  Si max<tab[i] max🡨tab[i]  I=i+1  Fintantque  Afficher (“le maximum est “,max) | var tab=[15,17,1,3]  var i=0  var max=tab[i]  while (i<4) {      if (max<tab[i]) max=tab[i]      i++  }  console.log('le maximum est: ',max) | | | | | var tab=[15,17,1,3,18,20,32,45,60]  function max(tableau){   var i=0   var max=tab[i]   while (i<tableau.length) {      if (max<tab[i]) max=tab[i]      i++   }   console.log('le maximum est: ',max)  }  max(tab) |

Mettre les données sous forme de tableau

* **Réaliser un programme qui va trier cet array dans l’ordre croissant**

|  |  |  |
| --- | --- | --- |
| Valeurs | Indice | Résultat |
| 17 | 0 | 15 |
| 15 | 1 |  |
| 1 | 2 |  |
| 3 | 3 |  |

On va comparer les chiffres sur 2 lignes et les intervertir si l’un est plus petit

Pour trier notre tableau, il faut relancer le code tant qu’il y a une interversion

|  |  |  |
| --- | --- | --- |
| Si tab(0)>tab(1) resultat🡨tab(1) ;tab(1)🡨tab(0) ;tab(0)🡨resultat;  *Si tab de 1 est supérieur à tab de 2 (si oui, stocker tab de 2 dans résultat) ;tab de 1 va être remplacer par tab 2 ;*  Si tab(1)>tab(2) resultat🡨tab(2) ;tab(2)🡨tab(1) ;tab(1)🡨resultat;  Si tab(2)>tab(3) resultat🡨tab(3) ;tab(3)🡨tab(2) ;tab(2)🡨resultat; | I🡨0  Tantque (i<tab.length)  Si tab(i)>tab(i+1)  Resultat🡨tab(i+1) ;tab(i+1)🡨tab(i) ;tab(i)🡨resultat;  I🡨i+1  fintantque | var tab=[15,1,3,17]  var i=0;  while (i<tab.length) {      if (tab[i]>tab[i+1]){          resultat=tab[i+1];          tab[i+1]=tab[i];          tab[i]=resultat;      }      i=i+1  }  console.log(tab) |