

**Exercice 1 :**

En algorithme	Implémentation en Python
<p><b>Procédure TermeN (n : entier)</b>  <b>Début</b>  <math>U \leftarrow 5</math>  <b>Pour</b> i de 1 à n-1 <b>Faire</b>  <math>U \leftarrow 2*U + 1.5</math>  Ecrire (U, " ")  <b>Fin Pour</b>  <b>Fin</b></p>	<pre>def TermeN (n) :     U=5     for i in range(1,n) :         U = 2*U + 1.5         print (U, " ")</pre>

**Exercice 2 :**

En algorithme	Implémentation en Python								
<p><b>Algorithme SomTermeW</b>  <b>Début</b>  L=Saisir ()  Chercher (L, Dt, W, Di)  Afficher (W, Di)  <b>Fin</b></p> <p><b>Fontion Chercher (L : entier)</b>  <b>Début</b>  <math>W[0] \leftarrow 7</math>  <math>Som \leftarrow 7</math>  <math>i \leftarrow -1</math>  <b>Tant que</b> <math>Som \leq L</math> <b>faire</b>  <math>i \leftarrow i+1</math>  <math>W[i] \leftarrow 2 * W[i-1] + 3.5</math>  <math>Som \leftarrow Som + W[i]</math>  <b>Fin Tant que</b>  <math>Dt \leftarrow W[i]</math>  <math>Di \leftarrow i</math>  retourner (W, Dt, Di)  <b>Fin</b></p> <p>▪ <b>TDO.Locaux</b></p> <table border="1"> <thead> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td><u>LDi</u></td> <td>Entier</td> </tr> <tr> <td><u>Som, Dt</u></td> <td>Réel</td> </tr> <tr> <td><u>W</u></td> <td>Tab</td> </tr> </tbody> </table>	Objet	Type/Nature	<u>LDi</u>	Entier	<u>Som, Dt</u>	Réel	<u>W</u>	Tab	<pre>from numpy import array W =array ([int ( )] * 20)  def saisir():     while (True) :         L =         int(input ("L="))         if (100&lt;=L&lt;=500) :             brea     kreturn (L)  def Chercher (L) :     W[0] = 7     Som = 7     i = -1     while Som ≤ L :         i = i + 1         W[i] = 2 * W[i-1] + 3.5         Som = Som + W[i]Dt = W[i] Di = i return (W, Dt, Di)  def afficher(W, Di) :     for i in range (Di+1) :         print (W[i], " ", end="")</pre>
Objet	Type/Nature								
<u>LDi</u>	Entier								
<u>Som, Dt</u>	Réel								
<u>W</u>	Tab								
	<p><b>Résultat d'exécution pour L=400</b></p> <pre>L=400 3   9   21   45   93   189   381  </pre>								

**Exercice 3**

En algorithmme	Implémentation en Python																								
<p><b>Algorithme RechercheMinMax</b></p> <p><b>Début</b></p> <p><math>n \leftarrow \text{saisir}()</math></p> <p><math>\text{remplirN}(M, n)</math></p> <p><math>\text{afficher}(M, n)</math></p> <p><b>Fin</b></p> <p><b>TDNT</b></p> <table border="1" data-bbox="236 562 707 674"> <tr> <td>Mat=tableau de 20 lignes * 20 colonnes des entiers</td> </tr> </table> <p>▪ <b>TDO.Globaux</b></p> <table border="1" data-bbox="236 736 564 949"> <tr> <td>n</td> <td>Entier</td> </tr> <tr> <td>M</td> <td>Mat</td> </tr> <tr> <td>saisir</td> <td>Fonction</td> </tr> <tr> <td>remplir</td> <td>Procédure</td> </tr> <tr> <td>afficher</td> <td>Procédure</td> </tr> </table> <p><b>Fonction <math>\text{remplirN}(M : \text{Mat}; n : \text{entier})</math></b></p> <p><b>Début</b></p> <p>Pour i de 0 à n-1 Faire</p> <p>    Pour j de 0 à i Faire</p> <p>        Si (i=j ou j=0) alors</p> <p>            <math>M[i, j] \leftarrow 1</math></p> <p>        Sinon</p> <p>            <math>M[i, j] \leftarrow M[i-1, j] + M[i-1, j-1]</math></p> <p>        Fin Si</p> <p>    Fin Pour</p> <p>Fin Pour</p> <p><b>Fin</b></p> <p><b>Procédure <math>\text{afficher}(M : \text{Mat}; n : \text{entier})</math></b></p> <p><b>Début</b></p> <p>Pour i de 0 à n-1 Faire</p> <p>    Pour j de 0 à n-1 Faire</p> <p>        Si <math>j \leq i</math> Alors</p> <p>            Ecrire (<math>M[i, j], " ", \text{end}=""</math>)</p> <p>        Fin Pour j</p> <p>    Ecrire ("<math>\backslash n</math>")</p> <p>Fin Pour</p> <p><b>Fin</b></p> <p>▪ <b>TDO.Locaux</b></p> <table border="1" data-bbox="248 1895 544 1957"> <tr> <td>Objet</td> <td>Type/Nature</td> </tr> <tr> <td>i, j</td> <td>Entiers</td> </tr> </table>	Mat=tableau de 20 lignes * 20 colonnes des entiers	n	Entier	M	Mat	saisir	Fonction	remplir	Procédure	afficher	Procédure	Objet	Type/Nature	i, j	Entiers	<pre> from numpy import array M =array ([[int ( )] * 20 for i in range(21)])  def saisir(): while True : n = int(input("n=")) if(2&lt;=n&lt;=20): break return(n) #Matrice carrée  def remplirN(M,n): for i in range(n) : for j in range(i+1) : if i==j or j==0 : M[i,j]=1 else: M[i,j] = M[i-1,j] + M[i-1,j-1]  def afficher (M,n) : for i in range(n) : for j in range(n) : if j&lt;=i : print(M[i,j]," ", end="") print("")  n=saisir() remplirN(M,n) afficher(M, n) </pre> <table border="1" data-bbox="842 1592 1417 1944"> <tr> <td>n=8</td> </tr> <tr> <td>1  </td> </tr> <tr> <td>1  1  </td> </tr> <tr> <td>1  2  1  </td> </tr> <tr> <td>1  3  3  1  </td> </tr> <tr> <td>1  4  6  4  1  </td> </tr> <tr> <td>1  5  10  10  5  1  </td> </tr> <tr> <td>1  6  15  20  15  6  1  </td> </tr> <tr> <td>1  7  21  35  35  21  7  1  </td> </tr> </table>	n=8	1	1  1	1  2  1	1  3  3  1	1  4  6  4  1	1  5  10  10  5  1	1  6  15  20  15  6  1	1  7  21  35  35  21  7  1
Mat=tableau de 20 lignes * 20 colonnes des entiers																									
n	Entier																								
M	Mat																								
saisir	Fonction																								
remplir	Procédure																								
afficher	Procédure																								
Objet	Type/Nature																								
i, j	Entiers																								
n=8																									
1																									
1  1																									
1  2  1																									
1  3  3  1																									
1  4  6  4  1																									
1  5  10  10  5  1																									
1  6  15  20  15  6  1																									
1  7  21  35  35  21  7  1																									

**Exercice 4**

En algorithmme	Implémentation en Python								
<p><b>Algorithme <u>FacteursPremiers</u></b></p> <p><b>Début</b>  n=Saisir (n)  RemplirFact (Fact, f, n)  Afficher (Fact, f)</p> <p><b>Fin</b></p> <ul style="list-style-type: none"> <li>▪ <b>TDNT</b></li> </ul> <table border="1" data-bbox="236 600 632 674"> <tr> <td><b>Tab</b>= tableau de 100 entiers</td> </tr> </table> <ul style="list-style-type: none"> <li>▪ <b><u>TDO.Globaux</u></b></li> </ul> <table border="1" data-bbox="236 712 619 824"> <tr> <td><b>n, f</b></td> <td><b>Entier</b></td> </tr> <tr> <td><b>Fact</b></td> <td><b>Tab</b></td> </tr> </table> <p><b>Fonction RemplirFact ( n : entier)</b></p> <p><b>Début</b>  i ← 2  f ← -1  <b>Répéter</b>      <b>Si</b> n Mod i = 0 <b>Alors</b>          n ← n Div i          f ← f+1          Fact[f] ← i      <b>Sinon</b>          i ← i+1      <b>Fin Si Jusqu'à</b> (n = 1)  <b>Retourner</b> (Fact, f)  <b>Fin</b></p> <p><b>Procédure Afficher (Fact :Tab ; f : entier )</b></p> <p><b>Début</b>  <b>Pour</b> i de 0 à (f-1) <b>Faire</b>      Ecrire (Fact[i], " *")  <b>Fin Pour</b>  Ecrire (Fact[f], " ")  <b>Fin</b></p>	<b>Tab</b> = tableau de 100 entiers	<b>n, f</b>	<b>Entier</b>	<b>Fact</b>	<b>Tab</b>	<pre> from numpy import array  Fact  =array ([int ( )] * 100)  def saisir():     while (True) :         n = int(input("n="))         if(4&lt;=n&lt;=100) :             break     return (n)  def RemplirFact(n) :     i = 2     f = -1     while True :         if n % i == 0 :             n = n // i             f = f + 1             Fact[f] = i         else :             i = i + 1         if (n==1):             break     return (Fact, f)  def Afficher (Fact , f) :     for i in range (f) :         print (Fact[i], " * ",end="")     print (Fact[f])  n=Saisir() Fact,f =RemplirFact(n) Afficher (Fact, f) </pre> <p><b>Résultat d'exécution pour n=120</b></p> <table border="1" data-bbox="901 450 1396 546"> <tr> <td> </td> </tr> </table> <p><b>Résultat d'exécution pour n=120</b></p> <table border="1" data-bbox="901 1783 1396 1908"> <tr> <td>n=120</td> </tr> <tr> <td>2 * 2 * 2 * 3 * 5</td> </tr> </table>		n=120	2 * 2 * 2 * 3 * 5
<b>Tab</b> = tableau de 100 entiers									
<b>n, f</b>	<b>Entier</b>								
<b>Fact</b>	<b>Tab</b>								
n=120									
2 * 2 * 2 * 3 * 5									