



Level: 1st year of computer science  
 Course: ADS1

## Typical solution TD/TP N°: 07

Academic year:2023/2024  
 Chapter 5 : Matrices & strings

### Exercise 1: TD/TP

Write a program that performs the addition of two real-type matrices of size m/n."

<pre> <b>Algorithm</b> addition <b>Var</b> i, j, M, N :integer     A[100][100],B[100][100],S[100][100]:     <b>array</b> [100] ofreal <b>begin</b> write("entrer les dimensions des matrices &lt;=100") read(M,N) write("entrer la 1ere matrice") <b>for</b> i&lt;=0 <b>to</b> M-1 <b>do</b> <b>for</b> j&lt;=0 <b>to</b> N-1 <b>do</b>     write("A["i","j"]=&gt;")     read(A[i][j]) <b>end for</b> <b>end for</b> write("entrer la 2eme matrice") <b>for</b> i&lt;=0 <b>to</b> M-1 <b>do</b> <b>for</b> j&lt;=0 <b>to</b> N-1 <b>do</b>     write("B["i","j"]=&gt;")     read(B[i][j]) <b>end for</b> <b>end for</b> <b>for</b> i&lt;=0 <b>to</b> M-1 <b>do</b> <b>for</b> j&lt;=0 <b>to</b> N-1 <b>do</b>     S[i][j] ← A[i][j]+ B[i][j] <b>end for</b> <b>end for</b> write("voici la somme") <b>for</b> i&lt;=0 <b>to</b> M-1 <b>do</b> <b>for</b> j&lt;=0 <b>to</b> N-1 <b>do</b>     write(S[i][j]) <b>end for</b> <b>end for</b> <b>end</b>                 </pre>	<pre> #include &lt;stdio.h&gt; <b>int</b> main(){ <b>int</b> i, j, M, N; <b>float</b> A[100][100],B[100][100],S[100][100]; printf("entrer les dimensions des matrices &lt;=100"); scanf("%d%d", &amp;M, &amp;N) ; printf("entrer la 1ere matrice\n"); <b>for</b> (i=0 ;i&lt;M ;i++)     <b>for</b> (j=0 ;j&lt;N ;j++){         printf("A[%d,%d]=&gt;",i, j);         scanf("%f", &amp;A[i][j]);     } printf("entrer la 2eme matrice\n"); <b>for</b> (i=0 ;i&lt;M ;i++)     <b>for</b> (j=0 ;j&lt;N ;j++){         printf("B[%d,%d]=&gt;",i, j);         scanf("%f", &amp; B[i][j]);     } <b>for</b> (i=0 ;i&lt;M ;i++)     <b>for</b> (j=0 ;j&lt;N ;j++)         S[i][j]= A[i][j]+ B[i][j]; printf("voici la somme ") ; <b>for</b> (i=0 ;i&lt;M ;i++){     <b>for</b> (j=0 ;j&lt;N ;j++)         printf("%.2f\t",S[i][j]);     printf("\n"); } <b>return</b> 0 ; }                 </pre>
---	---

### Exercise 2: TP

Write a program that calculates the sum of the elements on the main diagonal. This sum is called the trace of the matrix.

<pre> <b>Algorithm</b> trace <b>Var</b> i, j, N :entire     tr: reel     A[100]: <b>array</b> [100] ofreal <b>begin</b> write("entrer la taille de la matrice &lt;=100") read(N) write("entrer la matrice") <b>for</b> i&lt;=0 <b>to</b> N-1 <b>do</b> <b>for</b> j&lt;=0 <b>to</b> N-1 <b>do</b>     write("A["i","j"]=&gt;")                 </pre>	<pre> #include &lt;stdio.h&gt; <b>int</b> main(){ <b>int</b> i, j N; <b>float</b> tr, A[100][100]; printf("entrer la taille de la matrice &lt;=100"); scanf("%d", &amp;N) ; printf("entrer la matrice\n"); <b>for</b> (i=0 ;i&lt; N ;i++)     <b>for</b> (j=0 ;j&lt;N ;j++){         printf("A[%d,%d]=&gt;",i, j);         scanf("%f", &amp;A[i][j]);                 </pre>
---	--



<pre> read(A[i][j]) end for end for tr←0 for i←0 to N-1 do   tr← tr + A[i][i] end for write("la trace=", tr) end                     </pre>	<pre> } tr=0; for (i=0 ;i&lt; N ;i++)   tr+= A[i][i]; printf("la trace=%.2f ", tr) ; return 0 ; }                     </pre>
---	--

### Exercise 3: TD

Given a matrix A(M, N) of real numbers, write a program that calculates the sum of each column and saves them in an array T.

<pre> <b>Algorithm</b> somme_colonne <b>Var</b> i, j, M, N :integer       A: <b>array</b> [100] of real       T: <b>array</b> [100] [100] of real  <b>begin</b> write("enter matrix size &lt;=100") read(M,N) write("enter matrix ") <b>for</b> i←0 to M-1 <b>do</b> <b>for</b> j←0 to N-1 <b>do</b>   write("A["i","j"]=&gt;")   read(A[i][j]) <b>end for</b> <b>end for</b> <b>for</b> j←0 to N-1 <b>do</b> T[j]←0   <b>for</b> i←0 to M-1 <b>do</b>     T[j]← T[j]+ A[i][j]   <b>End for</b> <b>End for</b> write("here is the sum of the columns ") <b>for</b> i←0 to N-1 <b>do</b>   write(T[i]) <b>end for</b> <b>end</b>                     </pre>	<pre> #include &lt;stdio.h&gt; <b>int</b> main(){ <b>int</b> i, j, M, N; <b>float</b> A[100][100],T[100]; printf("enter matrix size &lt;=100"); scanf("%d%d", &amp;M, &amp;N) ; printf("enter matrix \n"); <b>for</b> (i=0 ;i&lt;M ;i++)   <b>for</b> (j=0 ;j&lt;N ;j++){     printf("A[%d,%d]=&gt;",i, j);     scanf("%f", &amp;A[i][j]);   } <b>for</b> (j=0 ;j&lt;N ;j++){   T[j]=0;   <b>for</b> (i=0 ;i&lt;M ;i++)     T[j]+= A[i][j]; } printf("here is the sum of the columns ") ; <b>for</b> (i=0 ;i&lt; N ;i++)   printf("%.2f\t",T[i]); <b>return</b> 0 ; }                     </pre>
--	--

### Exercise 4: TP

A symmetric matrix M of order n is a square matrix (number of rows = number of columns) that satisfies the following condition:  $M[i, j] = M[j, i]$  for all i and j.

Write a program that checks if a matrix is symmetric or not.

<pre> <b>Algorithm</b> Symetric <b>Var</b> i, j, N :integer       M: <b>array</b> [100][100] of real       isSymetrique : booleen  <b>begin</b> write("enter matrix size &lt;=100") read(N) write("enter matrix ") <b>for</b> i←0 to N-1 <b>do</b> <b>for</b> j←0 to N-1 <b>do</b>   write("M["i","j"]=&gt;")   read(M[i][j]) <b>end for</b> <b>end for</b>                     </pre>	<pre> #include &lt;stdio.h&gt; <b>int</b> main(){ <b>int</b> i, j, N, isSymetrique ; <b>float</b> M[100][100]; printf("enter matrix size &lt;=100"); scanf("%d", &amp;N) ; printf("enter matrix \n"); <b>for</b> (i=0 ;i&lt; N ;i++)   <b>for</b> (j=0 ;j&lt;N ;j++){     printf("M[%d,%d]=&gt;",i, j);     scanf("%f", &amp;M[i][j]);   } isSymetrique=1 ;                     </pre>
--	--



```

end for
isSymetrique ← true
for i←0 to N-2 do
for j←i+1 to N-1 do
    if M[i][j]≠ M[j][i] then
        isSymetrique ←false
    end if
end for
end for
if isSymetrique then
    write("the matrix is symmetrical ")
else
    write("the matrix is not symmetrical ")
end if
end
    
```

```

for (i=0 ;i< N ;i++)
    for (j=0 ;j<N ;j++)
        if(M[i][j] != M[j][i]) isSymetrique=0;
    }
if(isSymetrique)
    printf("the matrix is symmetrical ");
else
    printf("the matrix is symmetrical ");
return 0 ;
}
    
```

### Exercise 5: TD

Write a program that, given a string of characters S, displays whether S is a palindrome (symmetric) or not.

```

Algorithm palindrome
Var i, j :integer
    S: string
    isPalindrome : booleen
begin
write("enter a string ")
read(S)
i←0
j←strlen(S)-1
isPalindrome←true
while i<j and isPalindrome do
    if S[i]≠ S[j] then
        isPalindrome ←false
    end if
    i←i+1
    j←j-1
end while
if isPalindrome then
    write("the string is Palindrome ")
else
    write("the string is not Palindrome ")
end if
end
    
```

```

#include <stdio.h>
#include <string.h>
int main(){
int i, j, isPalindrome ;
char S[100];
printf("enter a string ");
gets(S);
i=0;
j=strlen(S)-1;
isPalindrome=1;
while(i<j && isPalindrome) {
    if (S[i] != S[j]) isPalindrome=0;
    i++;
    j--;
}
if(isPalindrome)
    printf("the string is Palindrome ");
else
    printf("the string is not Palindrome ");
return 0 ;
}
    
```

### Exercise 6: TD/TP

Write a program that removes all occurrences of a character in a string by shifting the rest to the left.

```

Algorithm supprime_occurrences
Var i, j :integer
    S: string
    x : character
begin
write("enter a string ")
read(S)
write("enter the character you want to
delete ")
read(x)
i←0
j←0
while S[i]≠'\0' do
    if S[i]≠ x then
        S[j]← S[i]
    
```

```

#include <stdio.h>
#include <string.h>
int main(){
int i, j;
char S[100],x;
printf("enter a string ");
gets(S);
printf("enter the character you want to
delete ");
x=getch();
i=0;
j=0;
while(S[i]!=='\0') {
    if (S[i] !=x) {
        S[j]= S[i];
    
```

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de M'sila  
Faculté des Mathématiques et de l'Informatique  
Département d'informatique



جامعة المسيلة  
كلية الرياضيات والإعلام الآلي  
قسم الإعلام الآلي

```
j←j+1
end if
i←i+1
end while
S[j]←'\0'
write("Here is the new channel ")
write(S)
end
```

```
j++;
}
i++;
}
S[j]='\0';
printf("Here is the new channel \n%s",S);
return 0 ;
}
```