

Correction of Exercise Series N°: 06

Ex1. Write an algorithm that reads an integer array of N elements, and displays the elements with odd index.

```
Algorithm display_odd_index
Var i, N :integer
      t:array [100] of real
begin
write("enter number of items <=100")
read(N)
for i←0 to N-1 do
    write(i,"=>")
    read(t[i])
end for
write("Here is the elements with odd indexs: ")
for i←0 to N-1 do
    if (i mod 2=1) then
        write(t[i])
    endIf
end for
end
```

Ex2. Write an algorithm that reads an array of N reals, it also reads an integer p, and then it puts the elements starting at index p into another array.

```
Algorithm New_array
Var i, N , p,j :integer
      t[100],V[100]:array of real
begin
write("enter number of items <=100")
read(N)
for i←0 to N-1 do
    write(i,"=>")
    read(t[i])
endfor
write("enter an index p : ")
read(p)
j←0
for i← p to N-1 do
    V[j] ←t[i]
    j←j+1
endfor
write("elements of the new array : ")
for i←0 to N-1 do
    write(V[i])
endfor
end
```

Ex3. Write an algorithm that reads a binary integer coded in SVA on 8 bits; the algorithm calculates its decimal value and its representation in C1 (index bit 0 is the sign bit)

```
Algorithm binary
Var i, N , p,j :integer
      SVA[8],C1[8]:array of integer
begin
for i←0 to 7 do
    write(i,"=>")
    read(t[i])
end for
D←0
P←2^6
for i← 1 to 7 do
    D ← D + t[i]*P
    P←P/2
endfor
if (t[0]=1) then
    D←-D
endIf
```

```

write("valeur decimal : ", D)
for i←1 to 7 do
  t[i] ← 1- t[i]
endfor
write("the number in C1 : ")
for i←0 to 7 do
  write(t[i])
endfor
end

```

Ex4. Write an algorithm that reads a matrix of R rows and C columns and displays the inner part of the matrix the inner part of the matrix → $i=0$ or $i=R-1$ or $j=0$ or $j=C-1$

2	4	9	1	9
3	7	5	2	1
8	1	2	1	3
4	7	8	6	2

inner part

```

Algorithm Inner_Part
Var i, j ,R,C:integer
      A[10][10]: array of real
begin
write("enter number of rows and number of columns : ")
read(R,C)
write("entrer la matrice")
for i←0 to R-1 do
for j←0 to C-1 do
  write("A["i",".",j,""]=>")
  read(A[i][j])
endfor
endfor
write("the inner part of matrix :")
for i←0 to R-1 do
for j←0 to C-1 do
  if (i≠0 and i≠R-1 and j≠0 and j≠C-1) then
    write (t[i][j])
  endif
endfor
endfor
end

```

Ex5.

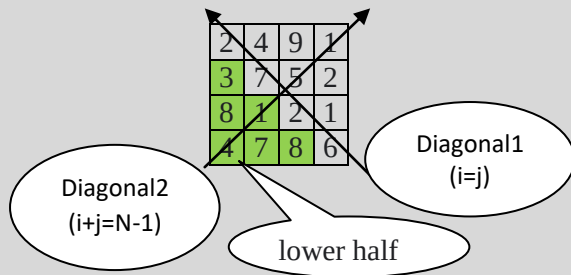
Write an algorithm that reads a matrix of real numbers, calculates and displays its transpose

```

Algorithm Transpose
Var i, j ,R,C:integer
      A[10][10],T[10][10]: array of real
begin
write("enter number of rows and number of columns : ")
read(R,C)
write("entrer la matrice")
for i←0 to R-1 do
for j←0 to C-1 do
  write("A["i",".",j,""]=>")
  read(A[i][j])
endfor
endfor
for i←0 to C-1 do
for j←0 to R-1 do
  T[i][j] ← A[j][i]
endfor
endfor
write ("the Transpose matrix :")
for i←0 to C-1 do
for j←0 to R-1 do
  write (T[i][j])
endfor
endfor
write ("\n")
endfor
end

```

Ex6. Write an algorithm that reads a square matrix and calculates the sum of its lower half and the sum of the two diagonals



```

Algorithm Transpose
Var i, j ,N,SumLow,Sumdiag1,SumDiag2:integer
      A[10][10],: array of real
begin
write("enter N : ")
read(N)
write("entrer the matrix elements: ")
for i←0 to N-1 do
for j←0 to N-1 do
    write("A["i",".",j,""]=>")
    read(A[i][j])
end for
end for
SumLow←0
SumDiag1←0
SumDiag2←0
for i←0 to N-1 do
for j←0 to N-1 do
    if (i<j) then
        SumLow←SumLow+A[i][j]
    Endif
    If (i=j) then
        SumDiag1=SumDiag1+A[i][j]
    endIf
    if (i+j=n-1) then
        SumDiag2←SumDiag2+A[i][j]
    endIf
end for
end for
    write ("Sum of lawer half : " ,SumLow)
    write ("Sum of diagonall : " ,SumDiag1)
    write ("Sum of diagonal2 : " ,SumDiag2)
end

```

Ex 7. Write an algorithm that reads a character string; it adds a period to the end of the string.

```

Algorithm palindrome
Var i, j :integer
      S: string
begin
write("enter a string : ")
read(S)
j←strlen(S)-1
S[j+1]='.'
S[j+1]=' \0'
write("Here is the new channel ")
write(S)
end

```

Ex 8. Write a program that reads a CH character string and converts all uppercase letters into lowercase letters and vice versa. The result will be stored in the same CH variable and displayed after the conversion.

```

Algorithm uptolow
Var i, j :integer
      S: string
begin
write("enter a string : ")
read(S)

```

```

i←0
while(S[i]≠ '\0')do
if (S[i]≥'a' and S[i]≤'z') then
    S[i] ←S[i]-32
Else
    If (S[i]≥'A' and S[i]≤'Z') then
        S[i] ←S[i]+32
    endIf
endIf
i←i+1
endWhile
write("Here is the new channel : ")
write(S)
end

```

Ex 9.Write an algorithm that reads a character string and starting from the character ';' it transfers the rest of the chain to another chain.

```

Algorithm transfer_string
Var i,j :integer
    S, T: string
begin
write("enter a string : ")
read(S)
i←0
while(S[i]≠ ';' and S[i]≠ '\0' )do
i←i+1
endWhile
if (S[i]=';' )then
    j←0
    while (S[i]≠ '\0')do
        T[j] ← S[i]
        i←i+1
        j←j+1
    endWhile
    T[j]='\0'
    write("Here is the new channel : ",T)
else
    write ("the char ; not exists")
endIf
End

```

Practical work

1. Write a program to calculate the number of occurrences of a given element in an integer array of N elements.

```

#include <stdio.h>
int main(){
int i, N ,x,Occ;
int t[100] ;
printf("enter number of items <=100");
scanf("%d", &N) ;
for ( i=0 ;i<N ;i++){
    printf("%d=>",i) ;
    scanf("%d", &t[i]) ;
}
printf("enter number : ");
scanf("%d", &x) ;
Occ=0;
for ( i=N-1 ;i>=0 ;i--)
    if(t[i]==x)
        Occ++;
printf("the number of occurrences of %d is %d ",x,Occ) ;
return 0 ;
}

```

2. Write a program that reads the averages of a module of n students; the program sets averages between 9.5 and 10 ; the program also displays a grade summary containing: best grade; bad grade ; students who have the module.

```
#include <stdio.h>
int main(){
int i, N ,adm;
float AVG[100], min, max ;
printf("enter number of students <=100");
scanf("%d", &N) ;
printf("enter the averages of students : ");

for ( i=0 ;i<N ;i++){
    printf("%d=>",i) ;
    scanf("%f", &t[i]) ;
}
for ( i=0 ;i<N ;i++)
if(AVG[i]>=9.5 && AVG[i]<10)
    AVG[i]=10;
min=AVG[0];
max=AVG[0];
adm=0;
for ( i=N-1 ;i>=0 ;i--)
{ if(AVG[i]>max)
max= AVG[i];
if(AVG[i]<min)
min= AVG[i];
if(AVG[i]>=10)
adm++;
}
printf("best grade:  %.2f \n ",max) ;
printf("bad grade:  %.2f \n ",max) ;
printf("number of students who have a module :
%d \n ",adm) ;
return 0 ;
}
```

3. Write a program that reads a matrix of R rows and C columns and sets its negative elements to zero.

```
#include <stdio.h>
int main(){
int i, j, R,C;
float tr, A[10][10];
printf("entrer number of rows and number of
columns <=100");
scanf("%d%d", &R,&C) ;
printf("entrer la matrice\n");
for (i=0 ;i< R ;i++){
    for (j=0 ;j<C ;j++){
        printf("A[%d.%d]=>",i, j);
        scanf("%f", &A[i][j]);
    }
}
for (i=0 ;i< R ;i++)
for (j=0 ;j<C ;j++)
    if( A[i][j]<0)
        A[i][j]=0;
printf("result matrix : \n");
for (i=0 ;i<R ;i++){
    for (j=0 ;j<C ;j++)
        printf("%d\t",A[i][j]);
    printf("\n");
}
return 0 ;
}
```

4. Write a program that reads a square matrix and check whether it is symmetric or not.

A symmetric matrix $\leftrightarrow M[i][j] = M[j][i] \forall i, j < n$

```
include <stdio.h>
int main(){
int i, j, N, isSymetrique ;
float M[100][100];
printf("enter matrix size <=100");
scanf("%d", &N) ;
printf("enter matrix \n");
for (i=0 ;i< N ;i++){
    for (j=0 ;j<N ;j++){
        printf("M[%d,%d]=>",i, j);
        scanf("%f", &M[i][j]);
    }
    isSymetrique=1 ;

    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 ;
}
```

5. Write a program that reads a CH character string and an integer n then deletes the last n characters of CH, finally it displays the CH string after this change.

```
#include <stdio.h>
#include <string.h>
int main(){
int i, j;
char S[100],x;
printf("enter a string ");
gets(S);
printf("enter a number n : ");
scanf("%d", &N) ;
i=strlen(s)-n;
S[i]= '\0';
printf("Here is the new channel \n%s",S);
return 0 ;
}
```

6. Write an algorithm that reads two character strings CH1 and CH2 and adds CH2 at the end of CH1.

```
#include <stdio.h>
#include <string.h>
int main(){
int i, j;
char CH1[100],CH2[100];
printf("enter the first string ");
gets(CH1);
printf("enter the second string ");
gets(CH2);
for (i=strlen(CH1),j=0 ;CH2[j]!='0';i++,j++)
    CH1[i]=CH2[j];
CH1[i]='\0';

printf("Here is the new channel \n%s",CH1);
return 0 ;
}
```