REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE 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

**Level**: 1st year of computer science
**Course**: ADS1

# Typical solution
# TD/TP N°: 06

**Academic year:**2023/2024
**Chapter 5** : Arrays

## Exercise 1: TD/TP

Write an algorithm with its C program that allows filling an array of N real numbers and then displays them in reverse order.

```
Algorithm inverse
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 reversed arra ")
for i←0 to N-1 do
  write(t[N-1-i])
end for
end
```

```
#include <stdio.h>
int main(){
int i, N ;
float t[100] ;
printf("enter number of items <=100");
scanf("%d", &N) ;
for ( i=0 ;i<N ;i++){
  printf("%d=>",i) ;
  scanf("%f", &t[i]) ;
}
printf("Here is the reversed arra ") ;
for ( i=N-1 ;i>=0 ;i--)
  printf("%.2f\t",t[i]) ;
return 0 ;
}
```

## Exercise 2: TP

Write an algorithm to endd the MAX and its position in an array of N real elements.

```
Algorithm inverse
Var i, N, p :integer
    t:array [100] of real
    MAX : real
begin
write("enter number of items <=100")
read(N)
for i←0 to N-1 do
  write(i,"=>")
  read(t[i])
end for
MAX←t[0]
p←0
for i←0 to N-1 do
  if t[i]>MAX then
    MAX← t[i]
    p← i
  end if
end for
write("the max is ", MAX, "sa position", p)
end
```

```
#include <stdio.h>
int main(){
int i, N , p;
float t[100] , MAX;
printf("enter number of items <=100");
scanf("%d", &N) ;
for ( i=0 ;i<N ;i++){
  printf("%d=>",i) ;
  scanf("%f", &t[i]) ;
}
MAX=t[0] ;
p=0 ;
for ( i=1 ;i<N ;i++)
  if(t[i]> MAX){
    MAX=t[i] ;
    p=i ;
  }
printf("the max is %.2f sa position %d",
MAX, p) ;
return 0 ;
}
```

## Exercise 3: TD

Write a C program to calculate the scalar product of two N-dimensional vectors. If u is the vector of grades and v is the vector of coefficients, modify the program to calculate the average.

```
Algorithm scalaire
Var i, N :integer
    u,v:array [100] of real
```

```
#include <stdio.h>
int main(){
int i, N;
```

```
    p: real
begin
write("enter number of items <=100")
read(N)
write("enter 1st vector ")
for i←0 to N-1 do
  write(i,"=>")
  read(u[i])
end for
write("enter 2nd vector ")
for i←0 to N-1 do
  write(i,"=>")
  read(v[i])
end for
p←0
for i←0 to N-1 do
  p←p+ u[i]* v[i]
end for
write("the scalar product ", p)
end
```

```
float u[100],v[100], p;
printf("enter number of items <=100");
scanf("%d", &N) ;
printf("enter 1st vector ");
for ( i=0 ;i<N ;i++){
  printf("%d=>",i) ;
  scanf("%f", &u[i]) ;
}
printf("enter 2nd vector ");
for ( i=0 ;i<N ;i++){
  printf("%d=>",i) ;
  scanf("%f", &v[i]) ;
}
p=0 ;
for ( i=0 ;i<N ;i++)
  p=p+ u[i]* v[i] ;
printf("the scalar product %f", p) ;
return 0 ;
}
```

```
Algorithm average
Var i, N, sc :integer
    u:array [100] of real
    v: array [100] of integer
    p, avg: real
begin
write("enter number of materials <=100")
read(N)
write("enter grade vector ")
for i←0 to N-1 do
  write(i,"=>")
  read(u[i])
end for
write("enter coefficient vector ")
for i←0 to N-1 do
  write(i,"=>")
  read(v[i])
end for
p←0
sc←0
for i←0 to N-1 do
  p←p+ u[i]* v[i]
  sc←sc +v[i]
end for
avg ←p/ sc
write("the average: ", avg)
end
```

```
#include <stdio.h>
int main(){
int i, N , sc,v[100];
float u[100], p, moy;
printf("enter number of items <=100");
scanf("%d", &N) ;
printf("enter grade vector ");
for ( i=0 ;i<N ;i++){
  printf("%d=>",i) ;
  scanf("%f", &u[i]) ;
}
printf("enter coefficient vector ");
for ( i=0 ;i<N ;i++){
  printf("%d=>",i) ;
  scanf("%f", &v[i]) ;
}
p=0 ;
sc=0 ;
for ( i=0 ;i<N ;i++){
  p=p+ u[i]* v[i] ;
  sc=sc + v[i] ;
}
avg =p/ sc ;
printf("the average: %f", avg) ;
return 0 ;
}
```

## Exercise 4: TP

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

```
Algorithm occurrences
Var i, N, nbO, x :integer
    t[100]:array [100] of integer
begin
write("enter number of items <=100")
read(N)
write("enter table elements ")
for i←0 to N-1 do
  write(i,"=>")
  read(t[i])
end for
write("enter nbr you are looking for ")
read(x)
```

```
#include <stdio.h>
int main(){
int i, N, nbO, x, t[100];
printf("enter number of items <=100");
scanf("%d", &N) ;
printf("enter table elements ");
for ( i=0 ;i<N ;i++){
  printf("%d=>",i) ;
  scanf("%d", &t[i]) ;
}
printf("enter nbr you are looking for ");
scanf("%d", &x) ;
nbO=0 ;
```

```
nbO←0
for i←0 to N-1 do
  if t[i]=x then
   nbO← nbO+1
  end if
end for
write(x " appeared ", nbO, "fois")
end
```

```
for ( i=0 ;i<N ;i++)
  if(t[i]==x) nbO++ ;
printf("%d appeared %d fois", x, nbO) ;
return 0 ;
}
```

### Exercise 5: TD

Write an algorithm with its C code that splits a vector T of N integers into two vectors T1 and T2, containing the even and odd numbers of T, respectively.

```
Algorithm split
Var i, j, k, N :integer
  t, t1, t2:array [100] of integer
begin
write("enter number of items <=100")
read(N)
write("enter table elements ")
for i←0 to N-1 do
  write(i,"=>")
  read(t[i])
end for
j←0
k←0
for i←0 to N-1 do
  if t[i] mod 2=0 then
   t1[j] ←t[i]
   j←j+1
  else
   t2[k] ←t[i]
   k←k+1
  end if
end for
write("here's the even table ")
for i←0 to j-1 do
  write(t1[i])
end for
write("here is the odd table ")
for i←0 to k-1 do
  write(t2[i])
end for
end
```

```
#include <stdio.h>
int main(){
int i, j, k, N, t[100] , t1[100] , t2[100];
printf("enter number of items <=100");
scanf("%d", &N) ;
printf("enter the table");
for ( i=0 ;i<N ;i++){
  printf("%d=>",i) ;
  scanf("%d", &t[i]) ;
}
j=0 ;
k=0 ;
for ( i=0 ;i<N ;i++)
  if(t[i]%2==0){
    t1[j]=t[i] ;
    j++ ;}
  else{
    t2[k]=t[i] ;
    k++ ;}
printf ("here's the even table ") ;
for ( i=0 ;i<j ;i++)
  printf("%d\t", t1[i]) ;
printf ("here is the odd table ") ;
for ( i=0 ;i<k ;i++)
  printf("%d\t", t2[i]) ;
return 0 ;
}
```

### Exercise 8 : ( at home)

A polynomial will be represented using an array of floating-point numbers containing its coefficients. The coefficient of degree i will be found in the array element at index i.

For example, the polynomial 2.5 + 4X + 8X^3 will be represented by the array [2.5, 4, 0, 8]. A polynomial of degree d will have a size of d+1.

Write an algorithm that:

- Reads the coefficients of the polynomial of degree d.
- Displays this polynomial, ignoring terms with coefficients of 0.
- Calculates the value of the polynomial for the given value x (provided by the user without using pow()).
- Calculates the derivative of this polynomial

```
#include <stdio.h>
int main(){
int i, d ;
float p[100] , dp[100],x,y;
printf("enter polynomial degree ");
scanf("%d", &d) ;
// reading the polynomial
printf("enter the coefficients of the polynomial ");
for ( i=0 ;i<=d ;i++){
  printf("x^%d " ,i) ;
  scanf("%f", &p[i]) ;
}
// polynomial display
for ( i=0 ;i<=d ;i++)
  if(p[i]<0)
      printf("%fx^%d" , p[i],i) ;
  else
   if(p[i]>0)
      printf("+%fx^%d" , p[i],i) ;
// calculates y=p(x)
printf("enter x");
scanf("%f", &x) ;
a=1 ;
y=0 ;
for ( i=0 ;i<=d ;i++){
  y=y+ p[i] *a ;
  a=a*x ;
}
printf ("y=%f\n",y) ;
// calculates the derivative of the polynomial
for ( i=0 ;i<d ;i++)
  dp[i]= p[i+1]*(i+1) ;
// derivative display
for ( i=0 ;i<d ;i++)
  if(dp[i]<0)
      printf("%fx^%d" , dp[i],i) ;
  else
   if(dp[i]>0)
      printf("+%fx^%d" ,dp[i],i) ;
return 0 ;
}
```