**UNIVERSITE DE BATNA**

## FACULTE DES SCIENCES DE L’INGENIEUR

# DEPARTEMENT DE MECANIQUE

MODULE : Programmation

DATE :11/04/2021

### SOLUTION DU CONTROLE EN DISTANCIEL

# Présentation (2.5 pts).

# EXERCICE N°1 (3.5 points)

Soit X un vecteur dont les N éléments sont donnés par :

# 

# 1) Développer un algorithme qui

# lit un entier N

# calcule et imprime les éléments du vecteur X.

2) Traduire cet algorithme en un programme Fortran.

|  |  |
| --- | --- |
| EXERCICE N°1 (3.5 points) **Algorithme** (2 points)  **Début**  **lire** n,x(1)  **pour** i=2 à n **faire**  x(i)**←**0  **pour** j=1 à i-1 **faire**  x(i)**←** x(i)+ x(j)  **fpr**  x(i)**←**(i-x(i))/i  **fpr**  **pour** i=1 à n **faire**  **write** x(i)  **fpr**  **fin** | **Programme** (1.5 points)  dimension x(30)  read(\*,\*) n, x(1)  do i=2, n  x(i)**=**0  do j=1, i-1  x(i)**=** x(i)+ x(j)  enddo  x(i)**=**(i-x(i))/i  enddo  do i=1, n  write(\*,\*)x(i)  enddo  stop  end |

# EXERCICE N°3 (3.5 points)

Considérons la somme suivante :



# Développer un algorithme qui calcule S

1. traduire cet algorithme en un programme Fortran

|  |  |
| --- | --- |
| EXERCICE N°3 (3.5 points) **Algorithme** (2 points)  **Début**  **lire** n,x  s**←** 0  **pour** i=1 à n **faire**  p**←**1  **pour** j=1 à n **faire**  **si** (j≠i) **alors**  p**←** p\*( x-j)/j  **fsi**  **fpr**  s**←**s**+**(1+i\*\*3)\*p  **fpr**  **écrire** s  **fin** | **Programme** (1.5 points)  read(\*,\*) n, x  s**=**0  do i=1, n  p=1  do j=1, n  if(j.ne.i) then  p=p\*( x-j)/j  endif  enddo  s=s**+**(1+i\*\*3)\*p  enddo  write(\*,\*)s  stop  end |

# EXERCICE N°4 (3.5points)



Rappelons qu’un polynôme peut être écrit sous la forme :

# .

# 1) En se basant sur ce rappel, développer un algorithme (organigramme) qui

# calcule , où est un nombre réel connu.

2) Traduire cet algorithme en un programme Fortran.

|  |  |
| --- | --- |
| EXERCICE N°4 (3.5 points) **Algorithme** (2 points)  **Début**  **lire** x,n  **pour** i=1 à n+1 **faire**  **lire** a(i)  **fpr**  z**←**a(n+1)  **pour** j=n à 1 avec un pas se -1 **faire**  z**←**z\*x+a(j)  **fpr**  **écrire** z  **fin** | **Programme** (1.5 points)  dimension a(20)  read(\*,\*)x,n  do i=1,n+1,1  read(\*,\*)a(i)  enddo  z=a(n+1)  do j=n, 1,-1  z=z\*x+a(j)  enddo  write(\*,\*)z  stop  end |

# EXERCICE N°5 (3.5 points)

# Développer un algorithme qui

# Lit un entier k

# calcule Sk=1-3+5-7+…+(-1)k-1(2k-1)

* imprime Sk.

# b) Traduire cet algorithme en un programme Fortran

|  |  |
| --- | --- |
| EXERCICE N°5(3.5 points) **Algorithme** (2 points)  **Début**  **lire** k  sk**←** 0  sgn**←**1  **pour** i=1 à k **faire**  sk**←**sk**+**sgn\*(2\*i-1)  sgn**←**- sgn  **fpr**  **écrire** sk  **fin** | **Programme** (1.5 points)  read(\*,\*) k  sk**=**0  sgn=1  do i=1, k  sk=sk**+**sgn\*(2\*i-1)  sgn=-sgn  enddo  write(\*,\*)sk  stop  end |

# EXERCICE N°6 (3.5 points)

Soit X un vecteur ayant n composantes.

# 1) Développer un algorithme qui

* Lit  n, et les composantes de X.
* Calcule et imprime la moyenne des composantes strictement positives, la moyenne des composantes négatives et le nombre des éléments nuls.

# 2) Traduire cet algorithme en un programme Fortran.

|  |  |
| --- | --- |
| EXERCICE N°6 (3.5 points) **Algorithme** (2 points)  **Début**  **lire** n  **pour** i=1 à n **faire**  **lire** x(i)  **fpr**  np**←**0  sp**←**0  nn**←**0  sn**←**0  **pour** i=1 à n **faire**  **si** (x(i)**.gt.**0) **alors**  np**←**np+1  sp**←**sp+x(i)  **sinonsi** (x(i)**.lt.**0)alors  nn**←**nn+1  sn**←**sn+x(i)  **fsi**  **fpr**  sp**←**sp/np  sn**←**sn/nn  n0**←**n-(np+nn)  **écrire** np,sp,nn,sn,n0  **fin** | **Programme** (1.5 points)  dimension x(30)  read(\*,\*) n  do i=1, n  read(\*,\*) x(i)  enddo  np=0  sp=0  nn=0  sn=0  do i=1, n  if(x(i) .gt.0) then  np=np+1  sp=sp+x(i)  elseif(x(i) .lt.0) then  nn**=**nn+1  sn**=**sn+x(i)  endif  enddo  sp=sp/np  sn=sn/nn  n0=n- (np**+** nn)  write(\*,\*)np,sp,nn,sn,n0  stop  end |