Exo1 Table de Multiplication

Procédure multiplication (n: entier) retourne void

variable

n, calcul:entier

g=string;

Début

Afficher(“ saisir n”)

Saisir(n);

calcul = 0;

for i dans 1..10 faire

calcul=i\*n;

Systeme.out.println(\n+i+”\*”+ n+” =”+ calcul);

fin for

fin Procédure

Exo2 Chiffre romains

1.

fonction chiffreUnites (nb : entier) retourne void

Action

// Pré-requis : 0 <= nb <= 9

// Action : affiche à l’écran la représentation de nb en chiffres

romains

Variable

nb=entier;

Romain=String;

Debut

saisir (nb);

If (nb<10) faire

if (nb<4)faire

For i dans 0..nb faire

Romain +=I;

fin for

else if (nb+1 mod 5 =0) faire

if (nb=4) faire

Romain+=IV;

else if (nb=9)

Romain+=IX;

else If (nb mod 5=0) faire

Romain +=V

if (nb> 5) faire

Romain+=V

For (0..nb-5) faire

Romain+=I;

fin for

fin if

fin if

fin if

fin if

fin if

fin if

fin if

2)

fonction chiffreDizaines (nb : entier) retourne void

Action

// Pré-requis : 0 <= nb <= 9

// Action : affiche à l’écran la représentation en chiffres romains de nb\*10.

Variable

nb=entier;

Romain=String;

Debut

saisir (nb);

If (nb<10) faire

if (nb<4)faire

For i dans 0..nb faire

Romain +=X;

fin for

else if (nb+1 mod 5 =0) faire

if (nb=4) faire

Romain+=XL;

else if (nb=9)

Romain+=XC;

else If (nb mod 5=0) faire

Romain +=X

if (nb> 5) faire

Romain+=L

For (0..nb-5) faire

Romain+=X;

fin for

fin if

fin if

3)

fonction chiffreCentaines (nb : entier) retourne void

Action

// Pré-requis : 0 <= nb <= 9

// Action : affiche à l’écran la représentation en chiffres romains de nb\*100.

Variable

nb=entier;

Romain=String;

Debut

saisir (nb);

If (nb<10) faire

if (nb<4)faire

For i dans 0..nb faire

Romain +=C;

fin for

else if (nb+1 mod 5 =0) faire

if (nb=4) faire

Romain+=CD;

else if (nb=9)

Romain+=CM;

else If (nb mod 5=0) faire

Romain +=D

if (nb> 5) faire

Romain+=D

For (0..nb-5) faire

Romain+=C;

fin for

fin if

fin if

3.B)

fonction chiffreMilliers (nb : entier) retourne void

Action

// Pré-requis : 0 <= nb <= 9

// Action : affiche à l’écran la représentation en chiffres romains de nb\*1000.

Variable

nb=entier;

Romain=String;

Debut

saisir (nb);

If (nb<10) faire

if (nb<5)faire

For i dans 0..nb faire

Romain +=M;

fin for

else if (nb=9)

Romain+=IX\*;

else If (nb mod 5=0) faire

Romain +=V\*

if (nb> 5) faire

Romain+=V;

For (0..nb-5) faire

Romain+=I\*;

fin for

fin if

fin if

5)

Procédure chiffresRomains (nb : entier) retourne void

Action

// Pré-requis : 0 <= nb <= 3999

// Action : saisit au clavier un entier compris entre 0 et 3999 et affiche à l’écran sa représentation en chiffres romains.

Variable

nb, quotient, copie=entier;

Romain=String;

Debut

afficher(“Sasir un nombre plus petit que 3999.”);

Saisir(nb);

Romains = “ “;

copie = nb;

while(copie.length !=1) faire

if (nb.length=2) faire

quotient =copie%10;

Romain+=chiffreDizaine(quotient);

copie%=10;

else if (nb.lenghth=3) faire

quotient =copie%10;

Romain+=chiffreCentaines(quotient);

copie%=10;

else if(nb.lenghth=4) faire

quotient =copie%10;

Romain+=chiffreMilliers(quotient);

copie%=10;

fin if

fin if

fin if

fin while

Romains +=chiffreUnites(copie);

Systeme.out.println(“nombre : “ + Romains);

fin procedure

exo5

fonction compactage (tableau v1) retourne un tableau t

Action

//renvoie un tableau compacté du tableau non vide donné

en paramètre

Variable

v2, tableau de v1.length()

Début

For i de 0.. v1.length() faire

if(v1[i]=0) faire

v2[i]=v1[i++];

v2[i++]=0;

fin if

fin for

fin Compactage

Exo6

fonction compactageSup (tableau v1) retourne un tableau t

Action

//on remplacera les valeurs significatives en double par des valeurs non significatives et on déplacera les valeurs significatives dans les éléments de faible indice du vecteur

Variable

v2, tableau de v1.length()

Début

For i de 0.. v1.length() faire

if(v1[i]=0) faire

v2[i]=v1[i++];

v2[i++]=0

fin if

fin for

fin Compactage

Compact et compact sup

java (en plusieur fonction)

public static int[] compactage(int[] t) {

int[] res = new int[t.length];

int indice = 0;

for(int i : t) {

if(i !=0) {

res[indice] = i;

indice++;

}

}

return res;

}

public static Boolean inTab(int[] tab, int val) {

int res = 0;

for(int i : tab) {

if(i == val) {

res = 1;

break;

}

}

return res == 1;

}

public static int[] compactageSup(int[] t) {

int indice = 0;

int[] res = new int[t.length];

for(int i : t) {

if(!inTab(res, i)) {

res[indice] = i;

indice++;

}

}

return res;

}