5.7 : le tir des nombres

Variable i,n : entiers

Début

Max :=0

Pour (i=a, i=20, i++) faire

Ecrire (« saisir le nombre »,i) ;

Lire (N) ;

\*Si (n>max) alors

Max :=n ;

Fin si

Fin pour

Ecrire (« le plus grand est », max)

Fin **PB SI IL N’Y A QUE DES NOMBRES NEGATIFS**

On n’initialise plus le max

* Si (i=1) alors max :=n ;
* Sinon si (n>max) alors

Max :=n ;

Fin si

Fin si

Encore plus simple : si (i=1) ou (n>max) alors max :=n ;

Pour la position : créer une nouvelle variable position :

\*Si (i=1) alors max :=n ;

Sinon si (n>max) alors

Max :=n ;

Position := i ;

Fin si

Fin si

Ecrire (« il est en position », position)

5.8 :

On ne sait pas combien de nombre saisi et saisie stop quand 0

Répéter jusqu’à car il faut écrire au moins une fois

Début

Variable i, n, position : entier ; done : boolen

I :=1 ;

done := faux ;

Répéter

Ecrire (« saisir le nombre », i) ;

Lire (n) ;

**Si (n=0) alors écrire (« au revoir ! ») ;**

**Sinon** Si (i=1 ou n>max) alors si (n!= 0 et (i=1 ou n>max))

Max : = n ;

Position : = i ;

Done := vrai

Fin si

**Fin si**

I++ ;

Jusqu’à (n=0) *(comparaison et non affectation)*

Si (done) alors

Ecrire (« Le max est », max, « il est apparu en position », position)

Fin

**Pb si personne saisi 0, le système va quand mm comparer et il affichera max = 0**

**Il y a toujours pb car mm si 0, affichera le message**

On ajoute donc un flag

5.9 :

Algorithme prix

Variable A, somme, rendu, Y, D, C, E : entiers

Début

somme : =0 *Initialisation*

Répéter

Écrire (« saisir prix ou 0 si plus d’articles »)

Lire (A)

Somme :=somme + A

Jusqu’à (A=0)

Ecrire (somme)

Ecrire (« Somme pour payer »)

Lire (Y)

Rendu= y-somme ;

D : = rendu/10 ;

C : = (rendu%10) /5 ;

E : = rendu%10 –C\*5 ;

Ecrire (« rendu de », D, « billets de 10€ », C, « billets de 5€ et », E, « pièces de 1€ ».)

Deuxième solution :

Rendu= y-somme ;

D : = rendu/10 ;

C : = (rendu%10) /5 ;

E : = rendu%5

Troisième solution : *constante sous chaine de caractères – defensive coding*

bD= « 10 euros », bC= « 5 euros », uE= « un euro »

écrire (d, bd, c, bc, e, ue)

5.10 :

Algorithme chevaux\_course

Variable x, y, n, p, pdta, pdtb, pdtc, i : entiers

Début

Ecrire (« saisir le nombre de chevaux en course »)

Lire (n)

Ecrire (« Nombre de chevaux joués »)

Lire (p)

Pdta=1

Pour (i=1, i<=n, i++) faire

Pdta : = pdta\*i ;

Fin pour

Pdtb : = 1 ;

Pour (i=1, i<= (n-p), i++) faire

Pdtb : =pdtb\*i ;

Fin pour

Pdtc : = 1 ;

Pour (i=1 ; i<=p ; i++) faire

Pdtc=pdtc\*i ;

Fin pour

X : = pdta/ pdtb

Y : = X /pdtc

Ecrire (« Dans le désordre : une chance sur », Y, « de gagner ») ;

Ecrire (« Dans l’ordre : une chance sur », X, « de gagner » ;

Fin

EXO DE COURS :

1)

DEBUT

*Ecrire (« saisir un nbr ») ;*

*Lire (A) ;*

I=0 ;

Jusqu’à i<11 ;

I :=i+1 ;

B := A \* i ;

*Ecrire (A, « x »,i, « = »,B) ;*

FIN

2)

4)

Algorithme table de multiplication

Variable i, A, produit : entiers

Début

I :=1 ;

A :=0 ;

Pour (i=1, i>=1,i++) faire

Pour

Tant que a<=10 faire

A := A+1 ;

Produit := i\*A ;

*Lire (produit) ;*

FIN

5)

DEBUT

*Ecrire (« saisir un nbr »)*

FIN

Algorithme minmax

Variable n, max, min, somme, moyenne

Done : = faux

Répéter

Ecrire (« nombre entre 0 et infini, sans virgule »,n) ;

Lire (ni) ;

Ecrire (« choisissez une consigne »)

Lire (consigne)

Done := faux

Somme := 0

Suivant consigne faire

« somme » : pour (i=1 ; i<=n ; i++) faire

somme := somme +ni;

fin pour

Ecrire (somme) ;

« moyenne » : pour (i=1 ; i<=n ; i++) faire

somme := somme +ni ;

Moyenne := somme/i ;

Fin pour

Ecrire (moyenne) ;

« min » : Si A<B

Ecrire (A, « est le minimum »)

Sinon écrire (B, « est le minimum »)

« max » : Si A>B

Ecrire (A, « est le maximum »)

Sinon écrire (B, « est le maximum »)

« quitter le programme » : done := vrai

Si (done) alors Fin

Fin

Exo bonus : pyramide de 99999

Début

Variable i : entiers

Début

Pour (i=1; i<=10, i++) faire

Pour (j=1, j<= 11-i, j++) faire

Ecrire (10-i) ;

Fin pour

Fin pour

Fin

**Correction :**

**Dbt**

**Pour (i :=10, i >=1,i--) faire**

**Pour (j=i, j>=1, j--) faire**

**Ecrire (i-1) ;**

**Fin pour**

**Ecrire (« \n ») ;**

**Fin pour**

**Fin**

QUESTION 4 :

Algorithme date\_valide

Variable : j,m,a, r, p, s : entiers positif

DEBUT

Ecrier (« saisir un jour »)

Lire (j)

Ecrier (« saisir un mois »)

Lire (m)

Ecrier (« saisir une année »)

Lire (a)

r := a%100 ;

p := a%4 ;

s := a%400 ;

Si (m>12 et j>31) alors

Ecrire (« date non valide ») ;

Sinon pour (m=1 ; m<=7 ;m+2) faire

écrire (« date valide ») ;

Fin pour

pour (m=8 ; m<=12 ;m+2) faire

écrire (« date valide ») ;

Fin pour

Sinon si (m = 4 ou m=6 ou m=9 ou m=11) **et** j<=30 alors

Ecrire (« date valide ») ;

Sinon si m := 2 et ((r !=0 et p=0) ou s=0) et j<=29 alors

Ecrire (« date valide») ;

Sinon si m := 2 et j<=28 alors écrire (« date valide ») ;

Sinon écrire (« date non valide ») ;

Fin si

Fin si

Fin si

Fin si

Fin si

Fin

**Correction**:

Si il y a que deux résultats, on cherche les exceptions

Il y a plus de résultats valides qu’invalides donc on va utiliser les cas de dates invalides dans le Si 🡪