EXERCICE 1

Ecrire un algorithme qui demande à l’utilisateur un nombre compris entre 1 et 3 jusqu’à ce que la réponse convienne.

Variable

nb est un entier

Debus de programmation

Nb🡨0

Debus tant que np<1 OU np>3 faire

Ecrire ‘’ saisir un nombre ‘’

Afficher nb

Fin de tant que

Ecrire ‘’ vous avez gagner ! ‘’

Fin de programmation

EXERCICE 2

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu’à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

Variable

Nb est un entier

Debus de programmation

Nb 🡨0

Debus de tant que np <10 OU np >20

Debus de si

Si np <10

Afficher ‘’ Plus grand !’’

Ecrire ‘’ saisir un nombre ‘’

Afficher np

Sinon

Afficher ‘’ Plus petit ! ‘’

Ecrire ‘’ saisir un nombre ‘’

Afficher np

Fin de si

Afficher ‘’ Vous avez gagne ‘’

Fin de programmation

Exercice avec un compteur !

Variable

nb est un entier

compteur est un entier

Debus de programmation

Nb 🡨0

Compteur prend 0

Debus de tant que np <10 OU np >20

Afficher ‘’Saisir un nombre’’

Ecrire nb

Compteur prend +1

Debus de si

Si np <10

Afficher ‘’ Plus grand !’’

Ecrire ‘’ saisir un nombre ‘’

Afficher np

Sinon

Afficher ‘’ Plus petit ! ‘’

Ecrire ‘’ saisir un nombre ‘’

Afficher np

Fin de si

Afficher ‘’ Vous avez gagné ‘’

Afficher ‘’ vous avez réussi’’ ,compteur

Fin de programmation

EXERCICE 3

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27. Ecrire cet algorithme dans un premier temps avec une boucle tant que, puis avec une boucle pour.

Pour tant que

Variable

np est un entier

i est un entier

Debus de programmation

np prend la valeur de 0

i prend la valeur de 0

afficher ‘’Saisir un nombre’’

écrire np

Debus de tant que

Tant que i <10

Np 🡨np+1

Afficher ,np

I 🡨i+1

Fin de tant que

Fin de programmation

Pour Pour

Variable

Np est un entier

I est un entier

Debus de programmation

Np🡨 0

Afficher ‘’ saisir un nombre ‘’

Ecrire np

Afficher ‘’Les valeurs des 10 nombres sont : ‘’

Debus de pour

Pour i allant de 1 a 10

Pour

Np 🡨+1

Afficher np

Fin de pour

Fin de programmation

EXERCICE 4

Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu’à ce nombre. Par exemple, si l’on entre 5, le programme doit calculer : 1 + 2 + 3 + 4 + 5 = 15 NB : on souhaite afficher uniquement le résultat, pas la décomposition du calcul.

Variable

nombre est un entier

i est un entier

resulta est un entier

debus de programmation

i🡨 0

resulta 🡨0

Afficher ‘’ saisir un nombre ‘’

Ecrire nombre

Debus de pour i=0 i<=nombre i++

Resulta 🡨resulta + i

Fin de pour

fin de programmation

EXERCICE 5 Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle. NB : la factorielle de 8, notée 8 !, vaut 1 x 2 x 3 x 4 x 5 x 6 x 7 x 8

Variable

nombre est un entier

i est un entier

resulta est un entier

debus de programmation

i🡨 1

resulta 🡨1

Afficher ‘’ saisir un nombre ‘’

Ecrire nombre

Debus de pour i=01 i<=nombre i++

Resulta 🡨resulta + i

Fin de pour

fin de programmation

EXRCICE 5

Un magasin de reprographie facture 0,10 € les dix premières photocopies, 0,09 € les vingt suivantes et 0,08 € au-delà. Ecrivez un algorithme qui demande à l’utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.

Variable

prix sont des réel

Nbphotocopies est un entier

Début de programmation

Afficher ‘’ Saisir le nombre de photocopies a faire

Ecrire nbphotocopies

Début de si

Si nbphotocopies <=10

Alors

Prix= nbphotocopies\*0.10

Afficher ‘’ le prix pour’’ , nbphotocopies ‘’ et ‘’ , prix

Sinon si nbphotocopies <= 30

Alors

Prix = (10\*0.10)+( nbphotocopies\*0.09)

Afficher ‘’ le prix pour’’ , nbphotocopies ‘’ et ‘’ , prix

Sinon

Prix = (10\*0.10)+(20\*0.09)+( nbphotocopies\*0.08)

Afficher ‘’ le prix pour’’ , nbphotocopies ‘’ et ‘’ , prix

Fin de programmation