**NOTIONS DE BASE D’UN ALGORITHME (LOGICIEL ALGOBOX).**

1. **VOLUME D’UNE SPHERE.**

***Il s’agit d’écrire un programme calculant le volume V d’une sphère de rayon R .***

On sait que : V = × π ×

1° Commencer par déclarer les variables : R : rayon, variable de type nombre

et : V : volume, variable de type nombre

2° Les 3 phases d’un algorithme :

1. Entrer les données. Ici **:  Lire la variable** – choisir le rayon R
2. Les traitements ou calculs : utiliser les boutons : **Nouvelle ligne** – puis **Affecter valeur à une variable** – choisir le volume V – puis dans la case « prend la valeur » écrire la formule (4/3)\*pow(R,3)\*Math.PI
3. L’affichage des résultats : utiliser : Nouvelle ligne- puis : **Ajouter afficher variable** – et choisir le volume V. L’affichage peut être amélioré si on ajoute au-dessus du texte avec **Ajouter Afficher Message.**

3° Pour tester l’algorithme, utiliser le bouton **Tester algorithme** puis : **Lancer algorithme**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Volume d’une sphère**  1 VARIABLES  2 R est du type Nombre  3 V est du type Nombre  4 DEBUT ALGORITHME  5 LIRE R  6 V prend la valeur (4/3)\*pow(R,3)\*Math.PI  7 AFFICHER V  8 FIN ALGORITHME. |  |  | | --- | | On utilise les 4 opérations : + ; - ; ; /  **Mais on met un point . pour les décimales ( et pas une virgule).** | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Fonction** | **Algobox** | |  | sqrt(x) | |  | pow(x,n) | | Nombre π | Math.PI | | Partie entière E(X) | floor(x) | | Reste de la division de x par y | x%y | | Arrondi à l’entier le plus proche | round(x) | | Obtenir un nombre aléatoire entre 0 et 1 | random | | cos x  sin x  tan x | cos(x)  sin (x)  tan (x | | ln x | log ( x )  exp ( x) | | |

1. **DISTANCE DE DEUX POINTS**

***On considère dans le plan rapporté à un repère orthonormé les deux points M1( X 1, Y1 ) et M2(X2 ,Y2) et on veut écrire un programme donnant la distance D=M1M2.***

On sait que : M1M2 = .

1° Commencer par déclarer les variables : X1, Y1, X2,Y2, puis X = X2 - X1 ,Y = Y2 - Y1 et enfin D (du type Nombre) . Il sera plus commode d’écrire : D = .

2° Les 3 phases d’un algorithme :

1. Lire les données : **Lire la variable**- choisir X1 . Puis faire de même avec Y1, X2, et Y2 .
2. Les traitements et calculs : utiliser : **Nouvelle ligne** puis **Affecter valeur à une variable** -choisir X et écrire X2 – X1 ; puis choisir Y et écrire Y2 – Y1 et enfin choisir D et écrire sqrt(pox(X,2)+pow(Y,2)).
3. L’affichage des résultats : **Nouvelle ligne**- puis : **Ajouter Afficher Variable** et choisir D.

3° Pour tester l’algorithme, utiliser le bouton : **Tester algorithme** , puis : **Lancer algorithme.**

|  |  |
| --- | --- |
| **DISTANCE DE 2 POINTS**  1 VARIABLES  2 X1 est du type Nombre  3 Y1 est du type Nombre  4 X2 est du type Nombre  5 Y2 est du type Nombre  6 X est du type Nombre  7 Y est du type Nombre  8 D est du type Nombre | 9 DEBUT ALGORITHME  10 Lire X1  11 Lire Y1  12 Lire X2  13 Lire Y2  14 X prend la valeur X2 – X1  15 Y prend la valeur Y2-Y1  16 D prend la valeur sqrt(pow(X,2)+pow(Y,2))  17 AFFICHER D  18 FIN ALGORITHME |

1. **INDICE DE MASSE CORPORELLE.**

On appelle « indice de masse corporelle » le nombre IMC = , où P est le poids en kilos, et T la taille en mètres. On convient que Si IMC<18,5, on est « MAIGRE » ; si 18,5 ≤ IMC < 25, on est « NORMAL » ; si 25 ≤ IMC <30, on est en « SURPOIDS » ; si IMC > 30, on est « OBESE ».

***Ecrire un programme calculant IMC (pour P et T donnés) et indiquant dans quelle catégorie on se place.***

**Mode conditionnel :**

Les instructions du type « Si…alors…sinon… »sont des **instructions conditionnelles.** La partie avec « sinon » est facultative. On peut imbriquer des instructions (faire « sinon si…. »).

Le bouton permettant d’ajouter du conditionnel dans Algobox est **« AJOUTER SI…ALORS** » et dans la fenêtre qui s’ouvre il faudra cocher la case **« AJOUTER SINON ».**

**Indications :**

1° Déclarer les variables P,T et IMC -du type Nombre

2° Lire P et T et utiliser « Affecter valeur à variable » pour écrire IMC= P/ pow(T,2). Faire afficher IMC.

3° Si IMC < 18.5 (attention c’est un point et pas une virgule pour les décimales) afficher le message « Maigre ». Sinon, si IM C< 25, afficher « Normal » ; sinon si IMC < 30 ,afficher « Surpoids » ; sinon afficher « Obèse ».

|  |  |
| --- | --- |
| **INDICE DE MASSE CORPORELLE**  1 VARIABLES  2 P EST DU TYPE NOMBRE  3 T EST DU TYPE NOMBRE  4 IMC EST DU TYPE NOMBRE  5 DEBUT ALGORITHME  6 LIRE P  7 LIRE T  8 IMC prend la valeur P/pow(T,2)  9 AFFICHER « valeur de IMC =  »  10 AFFICHER IMC  11 AFFICHER «  donc :  »  12 SI (IMC < 18.5) ALORS  13 DEBUT SI  14 AFFICHER « Maigre »  15 FIN SI  16 SINON  17 DEBUT SINON  18 SI (IMC<25) ALORS  19 DEBUT SI  20 AFFICHER « Normal »  21 FIN SI  22 SINON  23 DEBUT SINON  24 SI (IMC<30) ALORS  25 DEBUT SI  26 AFFICHER « Surpoids »  27 FIN SI  28 SINON  29 DEBUT SINON  30 AFFICHER « Obèse »  31 FIN SINON  32 FIN SINON  33 FIN SINON  34 FIN ALGORITHME | **JEU DU NOMBRE A TROUVER**  1 VARIABLES  2 A EST DU TYPE NOMBRE  3 N EST DU TYPE NOMBRE  4 I EST DU TYPE NOMBRE  5 T EST DU TYPE NOMBRE  6 DEBUT ALGORITHME  7 A prend la valeur round(random()\*100)+1  8 T prend la valeur 0  9 I prend la valeur 1  10 TANT QUE (T= =0) FAIRE  11 DEBUT TANT QUE  12 LIRE N  13 SI (N = = A) ALORS  14 DEBUT SI  15 T PREND LA VALEUR 1  16 AFFICHER « …Gagné en … »  17 AFFICHER I  18 AFFICHER « …essais… »  19 FIN SI  20 SINON  21 DEBUT SINON  22 SI (N <A) ALORS  23 DEBUT SI  24 AFFICHER « …Trop petit »  25 I prend la valeur I + 1  26 FIN SI  27 SINON  28 DEBUT SINON  29 AFFICHER « …Trop grand»  30 I prend la valeur I + 1  31 FIN SINON  32 FIN SINON  33 FIN TANT QUE  34 FIN ALGORITHME |

1. **JEU DU NOMBRE A TROUVER.**

***Le programme va choisir un entier entre 1 et 100 inclus et l’utilisateur va devoir le deviner le plus rapidement possible. A chaque essai le programme répond »Trop grand » ou « trop petit » ou encore « gagné en …essais ».***

**Comment obtenir des nombres aléatoires ?**

La fonction **random**  permet d’obtenir au hasard un nombre réel compris entre 0 et 1. Si on veut un entier ente 1 et 100 le nombre à deviner sera **A = round(random()\*100+1.**

**Faire une boucle :**

Algobox propose deux structures : **Ajouter POUR …DE…A** (faire pour I de 1 à N)

Ou **: Ajouter TANT QUE…**( Tant qu’une condition est vraie…)

**Tester une égalité.**

On pourra ici utiliser « Ajouter TANT QUE », avec une **variable test T :** continuer tant que le nombre A n’a pas été trouvé (s’il n’est pas trouvé la variable Test T vaudra 0 ; s’il est trouvé elle vaudra par exemple 1).

**Attention pour tester une égalité il faut deux fois le signe =** ( ici, écrire Tant que T = =0, et écrire si N = = A…).

**Indications :**

1° Déclarer les variables de type nombre A, nombre à deviner ; N : nombre entré par l’utilisateur  (qui change à chaque essai) ; I : variable comptant le nombre d’essais ; T : variable Test qui vaut 0 tant que l’utilisateur n’a pas trouvé le nombre A

2° Affecter à A la valeur round(random()\*100+1 , à T la valeur 0 et à I la valeur 1

3° Tant que T==0, lire N.

4° Si N==A, faire T=1, et afficher « gagné en » ; afficher I, puis afficher « essais ». Sinon, si N<A , afficher « Trop petit » et affecter à I la valeur I + 1 ; sinon afficher « Trop grand » et affecter à I la valeur I + 1 .

1. **UN JEU AVEC 5 DES**

***On lance 5 dés normaux dont les faces sont numérotés de 1 à 6. On gagne une partie lorsque la somme S est au moins égale à 20. Ecrire un programme qui permet de simuler ce lancer, de calculer la somme S obtenue et d’afficher si on a gagné ou perdu la partie.***

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **Un jeu avec 5 dés**  1 VARIABLES  2 I est du type Nombre  3 A est du type Nombre  4 S est du type Nombre  5 DEBUT ALGORITHME  6 S prend la valeur 0  7 POUR I ALLANT DE 1 A 5  8 DEBUT POUR  9 A prend la valeur round(random()\*6+1  10 S prend la valeur S+A  11 FIN POUR  12 AFFICHER S  13 SI (S>=20)ALORS  14 DEBUT SI  15 AFFICHER «  GAGNE »  16 FIN SI  17 SINON  18 DEBUT SINON  19 AFFICHER » PERDU »  20 FIN SINON  21 FIN ALGORITHME | | **Indications :**  1° Déclarer la variable numérique I, indice à faire varier de 1 à 5 (utiliser « Pour I allant de 1 à 5 »)  2° Déclarer la variable numérique A qui simule un lancer de dé : comme random permet d’obtenir un nombre aléatoire entre 0 et 1 , il suffit pour obtenir un nombre aléatoire compris entre 1 et 6 d’écrire A = round(random()\*6+1  3° Déclarer la variable numérique S qui calcule la somme des 5 nombres aléatoires obtenus et qu’il faudra initialiser à 0.  4° Faire afficher S et utiliser le test « Si S>=20) alors on a gagné (l’afficher) et sinon on a perdu (afficher le message).  **Suggestion**: répéter ce jeu 20 fois et compter le nombre de fois où on gagné… |