Algorithmique	Langage C	Python 3.6
Structure générale		
Début du programme Fin du programme	`int main()` `{` `return 0 ;` `}`	Aucune structure
Déclaration des variables		
i : entier	`int i ;`	Aucune déclaration : la déclaration d'une variable se fait simplement en lui affectant une valeur
rayon : réel	`float rayon ;`	
nom : chaine de caractère	`char nom[10]; // 10 caractères maxi`	
ma_lettre : caractère	`char ma_lettre; // 1 caractère`	
OuiNon : booléen		
Affectation d'une variable		
i <- 0	`i=0;`	`i = 0`
rayon <- 8.20	`rayon=8.2;`	`rayon = 8.2`
nom <- "Nestor"	`strcpy(nom,"Nestor");`[1][2]	`nom = "Nestor" # ou nom='Nestor'`
ma_lettre <- 'd'	`ma_lettre = 'd';`	N'existe pas
OuiNon <- VRAI		`OuiNon = True`
Afficher un message		
Afficher "Bonjour "	`printf("Bonjour\n")`[1]`;//avec saut de ligne` `printf("Bonjour"); // sans`	`print('Bonjour')# avec un saut de ligne` `print('Bonjour',end="");# sans`
Afficher "i=",i	`printf("i=%d\n",i);`	`print('i=',i)`
Afficher "rayon=",rayon	`printf("rayon=%f\n",rayon);`	`print("rayon=",rayon)`
Afficher "nom=", nom	`printf("nom=%s",nom)`	`print("nom=",nom)`
Afficher " ma_lettre =", ma_lettre	`printf("ma_lettre=%c\n",ma_lettre);`	
Afficher OuiNon		`print('OuiNon=',OuiNon)`
Affiche ""nom=", nom, " et le prénom=", nom	`printf("nom=%s et le prénom=%s",nom, prenom)`	`print("nom=",nom, "prénom=",nom)`

[1] `#include <stdio.h>`

[2] `#include <string.h>`

Saisir au clavier et affecter		
Demander nom	`scanf("%s",nom);`	`nom=input()`
Demander i	`scanf("%d",&i);`	`i=int(input())`
Demander rayon	`scanf("%f",&rayon);`	`rayon=float(input())`
Demander ma_lettre	`scanf("%c",&ma_lettre);`	N'existe pas
Commentaires		
Remarque	`// Sur une ligne` `/* Sur plusieurs` `lignes */`	`# Sur une ligne` `""" Sur plusieurs` `lignes """`
Actions conditionnelles		
Si (note>10) alors afficher « reçu » afficher « bravo » sinon afficher « recalé»	`if (note>10)` `{` `printf("reçu");` `printf("bravo");` `}` `else` `{` `printf("recalé");` `}`	`if note>10:` `print("reçu")#une tabulation avant` `print("bravo")` `else:` `print("recalé")`
Demander jour Selon jour : Cas 1 : afficher "lundi" Cas 2 : alors afficher "mardi" … Cas 7 : alors afficher "dimanche" Sinon : ("Erreur : le jour doit être compris entre 1 et 7") FinSelon	`int jour;` `printf("Entrez un jour de la semaine :");` `scanf("%d", &jour); printf("\n");` `switch (jour) {` `case 1 : printf("lundi\n"); break;` `case 2 : printf("mardi\n"); break;` `// et ainsi de suite` `case 7 : printf("dimanche\n"); break;` `default : printf("Erreur : le jour doit être compris entre 1 et 7\n");`	`#le switch case n'existe pas …..` `jour =int(input("Entrez un jour de la semaine :"))` `if (jour==1):` `print("Lundi")` `elif (jour==2):#mélange de else if` `print("Mardi")` `#et ainsi de suite` `else:` `print("Erreur : le jour doit être compris entre 1 et 7")`
Conditions booléennes		
Si (i=0) ET (rayon ≠0)	`if ((i==0) && (rayon!=0))`	`if (i==0) and (rayon!=0) :`
Si (aire≤10) OU (aire ≥20)	`if ((aire <=10.0) || (aire >=20.0))`	`if (aire <=10.0) or (aire>=20.0) :`
Si nom= "Nestor"	`if (strcmp(nom,"Nestor")==0)`[3]	`if nom=="Nestor" :`
Boucles		
Pour i variant de 1 à 10	`for( i = 1; i <=10; i=i+1) {` `printf("i=%d\n",i);`	`for i in range(1,11):` `print("i=",i)`

[3] `#include <string.h>`

afficher(i) Fp	}	
Tant que somme<100 faire somme=2*somme+1	`while (somme<100) {` `somme=2*somme+1;` `}` `printf("somme=%d\n",somme);`	`while somme<100:` `somme=2*somme+1` `print('somme=',somme)`
Reponse : caractère répéter lire reponse tant que reponse ≠ O	`do {` `printf("Voulez-vous sortir de la boucle (O/N) ?");` `scanf("%c", &reponse); printf("\n");` `}` `while (reponse != 'O');`	`while True:` `reponse = input("Voulez-vous sortir de la boucle (O/N) ? ")` `if reponse=="O":` `break`
Tableaux		
tab : tableau de 3 entiers tab[0]<-1 tab[1]<-10 tab[2]<-100 Afficher tab tab[0]<-2 tab[1]<-20 tab[2]<-200	`int tab[3]={1,10,100};` `for (int i = 0; i < 3; i++)` `printf("tab[%d]=%d ", i, tab[i]);` `// affiche tab[0]=1 tab[1]=10 tab[2]=100` `tab[0] = 2; // 1er élément` `tab[1] = 20; // 1er élément` `tab[2] = 200; //dernier élément` `for (int i = 0; i < 3; i++)` `printf("tab[%d]=%d ", i, tab[i]);` `// affiche tab[0]=2 tab[1]=20 tab[2]=200`	`tab=[1,10,100]` [4] `print(tab)` `#affiche [1, 10, 100]` `tab[0]=2` `tab[1]=20` `tab[2]=200` `print(tab)` `#affiche [2, 20, 200]`
tableau2D : tableau de 4 sur 3 entiers Afficher tableau2D	`int tab2D[4][3]={{0,0,0},{1,1,1},{2,2,2},{3,3,3}};// 4lignes*3colonnes` `for (int ligne = 0; ligne < 4; ligne++)` `for (int col = 0; col < 3; col++)` `printf("tab2D[%d][%d]=%d\n ", ligne,col, tab2D[ligne][col]);` `// affiche tab2D[0][0]=0 tab2D[0][1]=0 tab2D[0][2]=0 tab2D[1][0]=1 ...`	`tab2D=[[0,0,0],[1,1,1],[2,2,2],[3,3,3]]` `print(tab2D)` `#affiche [[0, 0, 0], [1, 1, 1], [2, 2, 2], [3, 3, 3]]`
noms : tableau de chaînes contenant "Alice", "Bobbii", "Samia", et "Zheng-You" Afficher noms[1]	`char noms[][16] = {"Alice", "Bobbie", "Samia", "Zheng-You"};` `printf("%s",noms[1]); // Affiche Bobbie`	`noms = ["Alice", "Bobbie", "Samia", "Zheng-You"]` `print(noms[1])` `#affiche Bobbie`
Fonctions		
Les procédures (fonction sans retour)		

[4] Utilisation des listes concept bien plus puissant (avec des fonctions comme tab.append pour ajouter à la fin, len(tab) longueur, del(tab[1]) supprimer …)

Définir procédure tirerUnTrait Aller à la ligne Afficher "------------------------------" Aller à la ligne	void tirerUnTrait()[5]{ println("\n----------\n"); } L'appel se fait dans le programme principal : tirerUnTrait() ;	def tirerUnTrait():[6] print("----------------") tirerUnTrait()
Fonctions avec arguments et valeur de retour		
Fonction abs Entrée: x: entier Sortie: entier Début Si (x > 0) Alors Retourner x Sinon Retourner -x Fin_abs Afficher("Entrer y: ") Saisir(y) Affichager abs(y)	int abs(int x) { if (x > 0) { return x; } else { return - x; } } int y; printf("Entrer y: "); scanf("%d", &y); printf("%d\n", abs(y));	def abs(x): if x>0: return x else: return -x y=int(input("Entrer y: ")) print(abs(y)) *#affiche par ex 2 si y=-2*
Opérations diverses		
Afficher ("13 modulo 5 = " 13 modulo 5) Le module est le reste de la division entière	printf("13 modulo 5 = %d",13%5); // affiche 3	print("13 modulo 5 = ",13%5)
Afficher ("un nombre aléatoire entre 0 et 10") Afficher ("un nombre aléatoire entre 1 et 10")	srand (time(NULL)); // indispensable[7] printf("nombre aléatoire entre 0 et 10 : %d ", rand() % 11); printf("nombre aléatoire entre 1 et 10 : %d ", rand() % 10 + 1);	import random print("nombre aleatoire entre 0 et 10 :" , random.randint(0, 10)) print("nombre aleatoire entre 1 et 10 :" , random.randint(1, 10))

1

[5] La déclaration se fait avant le main()

[6] La déclaration peut se faire au milieu du code, il est vivement conseillé de le mettre en tout début

[7] L'initialisation du générateur de nombres aléatoires, obligatoire sinon on aura toujours le même nombre aléatoire à chaque exécution du programme