

EN TÊTE

| | |
|-------------------------------------|---|
| #include <stdio.h> | Obligatoire |
| #include <stdlib.h> | Obligatoire |
| #include <string.h> | Obligatoire |
| #include <math.h> | Inclue la bibliothèque math (pour les formules) |
| #include <time.h> | Pour les nombres aléatoires |
| #include <ctype.h> | Pour les transformation de lettre(minuscule->majuscule) |
| | |
| #define nom () nom2 \ | Fonctionne comme une variable sans prendre de mémoire, remplace le nom par nom2 Le \ permet de cree une 2 ^e ligne tout en restant dans le meme define |

CORP

| | |
|---|--|
| int main() | Fonction Principal |
| { | |
| printf("..."); | Affiche le message ... |
| //... ou *...*\ | Commentaire invisible |
| printf("... %d / %ld / %f / %f / %c ..., nom_variable "); | Choisi le type de variable et son nom %d = int %ld = long %f = float ou double %s = chaine caractere de char %c = lit dernière lettre mot |
| Scanf("%d / %s", & nom variable) | L'utilisateur entre le nombre(%d) ou la chaine de caractere [%s] de la variable |
| While(...) | Tant que ... |
| {...} | Execute ... |
| For(...) | pour ... |
| {...} | Execute ... |
| System("cls"); | Vide la console |
| sprintf(char_a_ecrire_dedans , "...%d...", variable de %d); | Ecrit dans char_a_ecrire_dedans , le texte ...%... |
| | |


Dans les messages (printf)

| | |
|----|-------------------|
| \n | Retour à la ligne |
| \t | Tabulation |

Lire/Ecrire dans un fichier

| | |
|---|--|
| FILE* fichier = NULL; | Cree un pointeur fichier permettant la lecture/écriture |
| fichier = fopen(" nom .txt", " type ") | Ouvre le fichier nom et le met à un type adapter r : lecture seul w : écriture seul (créé le fichier s'il n'existe pas) a : mode d'ajout (écrit à partir de la fin) r+ : lecture et écriture w+ : suppression du contenu puis écriture/lecture a+ : écriture et lecture à partir de la fin Mettre une condition " ficher != NULL ", après cette commande |
| - fputc('A', fichier) - fputs(" phrase ", fichier) - fprintf(fichier , " phrase ", variable) | - Ecrit la lettre A dans fichier - Ecrit une phrase (possible d'aller à la ligne) dans fichier - Ecrit dans fichier la phrase en utilisant la variable |
| Variable= fgetc(fichier) fgets(char , nombre_caractère , fichier); While(fgets(char , nombre_caractère , fichier)) != NULL fscanf(fichier , "%d %d %d", & variable [0], & variable [1], & variable [2]); | - Lit un à un le caractère du fichier et l'insère dans variable (à chaque relance, lit le caractère suivant) [EOF = fin de fichier, pour le "tant que"] - Lit une ligne (maximum) du fichier et l'insère dans char (à chaque relance, lit la ligne suivante) - Lit tout le fichier - Met dans chaque variable le %d du fichier , une variable = un mot/chiffre seulement. (exemple contenue fichier = 15 20 30, variable[0] = 15, variable[1] = 20, variable[2] = 30) |
| rename(" fichier ", " fichier2 ") | Renome le fichier en fichier2 |
| remove(" fichier ") | Supprime le fichier |
| fclose(fichier); | Ferme le fichier ouvert |

VARIABLES

| | |
|--|--|
| char ... = '...'; | Permet d'enregistré UN caractere dans CCC |
| Short ...; | -32 768 32 767 |
| Int ...; | -32 768 32 767 |
| Long ...; | -2 147 483 648 2 147 483 647 |
| Long long ...; | -9 223 372 036 854 775 808 9 223 372 036 854 775 807 |
| Float ...; | -1038 1037 |
| Double ...; | -1038 1037 |
| Unsigned type | Accepte que le positif |
| Const type | Impossible de modifié la variable |
| Static type | Le variable fonctionne que dans ce fichier |
| nom_variable = fabs(nombre) | Met le négatif en positif (fabs pour double, abs pour int) |
| nom_variable = abs(nombre) | |
| nom_variable = ceil(nombre) | Arrondi au-dessus (53,7 devient 54) |
| nom_variable = floor(nombre) | Arrondi au-dessous (53,7 devient 53) |
| nom_variable = pow(nombre, puissance) | Calcul la puissance |
| nom_variable = sqrt(nombre) | Calcul la racine carré |
| nom_variable = sqrt(nombre) | |
| nom_variable = sin / cos / tan / asin / acos / atan / exp / log / log10 (nombre) | Sinus, cosinus, tangente, arc sinus, arc cosinus, arc tangente, exponentiel, logarithme, logarithme base 10 |
| srand(time(NULL)); | Srand à mettre en début de programme |
| nom_variable = (rand() % (MAX - MIN + 1)); | Nombre aléatoire entre MAX et MIN |
| nom_variable[chiffre] | Cree un tableau de chiffre case, sur 4 adresses se suivant (1600/1601/1602/1603) |
| caractere = getchar() | Lit le caractère (entré par l'utilisateur) |
| Nom_variable1 = strlen(nom_variable_char) | Calcul le nombre de caractère de la variable CHAR |
| strcpy(copieDeLaChaine, chaineACopier); | Permet de copier une chaine de caractère chaineACopier dans une autre chaine de caractère copieDeLaChaine |
| strcat(chaine1, chaine2) | Concacte les 2 chaine dans chaine1 Chaine1 = bonjour }  chaine1 devient : bonjour Chaine2 = alexandre } alexandre |
| Pointeur = strchr(chaine, 'lettre') | Le pointeur est égal à la fin de chaine à partir de la première lettre trouvé |
| Pointeur = strchr(chaine, 'lettre') | Idem mais à partir de la dernière lettre trouvé |
| Pointeur = strpbrk("chaine", "lettrelettrelettre..."); | Le pointeur est égal à la fin de chaine à partir de la première lettre trouvé entre celles définie |
| Pointeur = strstr("chaine", "chaine_recherché"); | Le pointeur est égal à la fin de chaine à partir du mot chaine_recherché |
| Variable = toupper(Variable) | Met variable en majuscule |

CONDITIONS

| | |
|---|--|
| If (...) {...} | Si ... exécute ... |
| If (!(...)) {...} | Si ... n'est pas, exécute ... |
| Elseif (...) {...} | Sinon si ... exécute ... |
| Else {...} | Sinon exécute... |
| Dans if : && / | Et / ou |
| Switch (nombre) {case ...: ... Breack; Case...: ... Breack; Default: ... Breack; | Si nombre = ... Exécute ... STOP switch Si nombre = ... Exécute ... STOP switch Si aucun des nombres n'est dans la liste des cases Exécute ... STOP switch |

PROTOTYPES

| | |
|---|--|
| Int nom (int nom variable) {code} | Crée un prototype (en dehors de int main) et la variable pour return Code C du prototype |
| Nom (nom variable); | Appelle le prototype avec la variable qu'on veut |
| Return nom variable | Retourne la valeur de la variable du prototype |
| | |
| int * nom = NULL; | Créer un pointeur nom |
| Nom = & nom2 | Le pointeur nom prend l'adresse de la variable nom2 , dans "main" demander * nom pour avoir la valeur de nom2 |
| Dans main : nom_prototype (& nom_variable1 , & nom_variable2 ,...); void nom_prototype (int * nom_pointeur1 , int * nom_pointeur2 , ...) {code} | On appelle le prototype avec les variables dont on veut modifier On remet dans l'ordre les pointeurs lié aux variables (dans prototype) |