

<pre>#include <stdio.h> int main() { int n, i; unsigned long long factorielle = 1; printf("Entrez un nombre entier positif : "); scanf("%d", &n); if (n < 0) { printf("La factorielle n'est pas définie pour les nombres négatifs.\n"); } else { for (i = 1; i <= n; ++i) { factorielle *= i; } printf("Factorielle de %d = %llu\n", n, factorielle); } return 0; }</pre>	<pre>#include <stdio.h> int main() { int nombres[5] = {10, 20, 30, 40, 50}; int i; printf("Les nombres dans le tableau sont : "); for (i = 0; i < 5; i++) { printf("%d ", nombres[i]); } printf("\n"); return 0; }</pre>	<pre>#include <stdio.h> #include <string.h> // Inclure string.h pour les fonctions sur les chaînes int main() { char chaine[50] = "Bonjour"; int longueur; longueur = strlen(chaine); // Calcule la longueur de la chaîne printf("La chaîne est : %s\n", chaine); printf("Sa longueur est : %d\n", longueur); strcat(chaine, " le monde !"); // Concatène des chaînes printf("La chaîne concaténée est : %s\n", chaine); return 0; }</pre>
<pre>#include <stdio.h> // Déclaration de la fonction int addition(int a, int b); int main() { int x = 5, y = 3; int somme = addition(x, y); // Appel de la fonction printf("La somme de %d et %d est : %d\n", x, y, somme); return 0; } // Définition de la fonction int addition(int a, int b) { return a + b; } ***** #include <stdio.h> #include <stdlib.h> // Pour exit() int main() { int choix; float nombre1, nombre2, resultat; char operateur; do { printf("\nMenu :\n"); printf("1. Addition (+)\n"); printf("2. Soustraction (-)\n"); printf("3. Multiplication (*)\n");</pre>	<pre> printf("4. Division (/)\n"); printf("0. Quitter\n"); printf("Votre choix : "); if (scanf("%d", &choix) != 1) { printf("Erreur de saisie. Veuillez entrer un nombre.\n"); // Nettoyer le buffer d'entrée pour éviter une boucle infinie while (getchar() != '\n'); continue; // Retour au début de la boucle } if (choix == 0) break; // Quitter la boucle si l'utilisateur choisit 0 printf("Entrez deux nombres : "); if (scanf("%f %f", &nombre1, &nombre2) != 2) { printf("Erreur de saisie. Veuillez entrer des nombres.\n"); while (getchar() != '\n'); continue; } switch (choix) { case 1: operateur = '+'; resultat = nombre1 + nombre2; break; case 2: operateur = '-'; resultat = nombre1 - nombre2; break;</pre>	<pre> case 3: operateur = '*'; resultat = nombre1 * nombre2; break; case 4: if (nombre2 == 0) { printf("Erreur : Division par zéro impossible.\n"); continue; } operateur = '/'; resultat = nombre1 / nombre2; break; default: printf("Choix invalide.\n"); continue; } printf("%.2f %c %.2f = %.2f\n", nombre1, operateur, nombre2, resultat); } while (1); // Boucle infinie jusqu'à ce que l'utilisateur choisisse de quitter printf("Programme terminé.\n"); return 0; }</pre>

<pre> #include <stdio.h> #include <stdlib.h> #include <time.h> int main() { int nombre_secret, proposition, tentatives = 0, choix; srand(time(NULL)); // Initialisation du générateur de nombres aléatoires nombre_secret = (rand() % 100) + 1; do { printf("\nMenu :\n"); printf("1. Proposer un nombre\n"); printf("2. Abandonner\n"); printf("Votre choix : "); if (scanf("%d", &choix) != 1){ printf("Erreur de saisie. Veuillez entrer un nombre.\n"); while (getchar() != '\n'); continue; } switch (choix) { case 1: printf("Entrez votre proposition (entre 1 et 100) : "); if (scanf("%d", &proposition) != 1){ printf("Erreur de saisie. Veuillez entrer un nombre.\n"); while (getchar() != '\n'); continue; } tentatives++; if (proposition < nombre_secret) { printf("C'est plus grand.\n"); } else if (proposition > nombre_secret) { printf("C'est plus petit.\n"); } else { printf("Bravo ! Vous avez trouvé le nombre en %d tentatives.\n", tentatives); return 0; // Fin du jeu } break; case 2: printf("Vous avez abandonné. Le nombre était %d.\n", nombre_secret); return 0; // Fin du jeu default: printf("Choix invalide.\n"); } } while (1);return 0; } </pre>	<pre> #include <stdio.h> #include <string.h> #include <stdbool.h> // Pour utiliser le type booléen #define MAX_TACHES 10 #define MAX_NOM_TACHE 50 int main() { char taches[MAX_TACHES][MAX_NOM_TACHE]; bool taches_terminees[MAX_TACHES] = {false}; // Tableau pour suivre l'état des tâches int nombre_taches = 0; int choix; do { printf("\nGestionnaire de tâches :\n"); printf("1. Ajouter une tâche\n"); printf("2. Afficher les tâches\n"); printf("3. Marquer une tâche comme terminée\n"); printf("4. Quitter\n"); printf("Votre choix : "); if (scanf("%d", &choix) != 1) { printf("Erreur de saisie. Veuillez entrer un nombre.\n"); while (getchar() != '\n'); continue; } switch (choix) { case 1: if (nombre_taches < MAX_TACHES) { printf("Entrez le nom de la tâche : "); // Nettoyer le buffer avant de lire la chaîne while (getchar() != '\n'); if (fgets(taches[nombre_taches], MAX_NOM_TACHE, stdin) != NULL) { // Supprimer le caractère de nouvelle ligne ajouté par fgets taches[nombre_taches][strcspn(taches[nombre_taches], "\n")] = 0; nombre_taches++; printf("Tâche ajoutée.\n"); } } } } </pre>	<pre> } else { printf("Erreur lors de la lecture de la tâche.\n"); } } else { printf("La liste de tâches est pleine.\n"); } break; case 2: if (nombre_taches == 0) { printf("Aucune tâche à afficher.\n"); } else { printf("Liste des tâches :\n"); for (int i = 0; i < nombre_taches; i++) { printf("%d. [%c] %s\n", i + 1, taches_terminees[i] ? 'X' : ' ', taches[i]); } } break; case 3: if (nombre_taches == 0) { printf("Aucune tâche à marquer.\n"); } else { printf("Numéro de la tâche à marquer comme terminée : "); int num_tache; if (scanf("%d", &num_tache) != 1) { printf("Erreur de saisie. Veuillez entrer un nombre.\n"); while (getchar() != '\n'); continue; } if (num_tache > 0 && num_tache <= nombre_taches) { taches_terminees[num_tache - 1] = true; printf("Tâche marquée comme terminée.\n"); } else { printf("Numéro de tâche invalide.\n"); } } break; case 4: printf("Programme terminé.\n"); break; default: printf("Choix invalide.\n"); } while (choix != 4); return 0; } </pre>
--	---	--