

```

# Début du programme

# Boucle principale pour permettre à l'utilisateur de continuer ou de quitter
while True:
    # Boucle pour demander un nombre entier positif à l'utilisateur
    while True:
        try:
            # Demande à l'utilisateur d'entrer un nombre entier positif
            nombre = int(input("Entrez un nombre entier positif : "))
            # Vérifie si le nombre est positif
            if nombre > 0:
                break # Si le nombre est positif, sortir de la boucle
            else:
                print("Erreur : Veuillez entrer un nombre entier positif.")
        except ValueError:
            # Gère le cas où l'utilisateur n'entre pas un nombre entier
            print("Erreur : Veuillez entrer un nombre entier valide.")

# Affichage de la table de multiplication
print(f"\nTable de multiplication de {nombre} :")
# Boucle pour générer les lignes de la table de multiplication
for i in range(1, 11):
    # Calcul et affichage de chaque ligne
    print(f"{nombre} x {i} = {nombre * i}")

# Demande à l'utilisateur s'il veut entrer un autre nombre ou quitter
recommencer = input("\nVoulez-vous entrer un autre nombre ? (oui/non) : ").strip().
if recommencer != 'oui':
    # Si l'utilisateur ne veut pas recommencer, sortir de la boucle principale
    print("Au revoir !")
    break # Utilisation du mot-clé 'break' pour sortir de la boucle principale
else:
    # Si l'utilisateur veut recommencer, continuer la boucle principale
    continue # Utilisation du mot-clé 'continue' pour recommencer la boucle princi

```

Explications détaillées :

1. Boucle principale avec `while True` :

- La boucle principale permet de continuer à demander un nombre ou de quitter le programme selon le choix de l'utilisateur.

2. Boucle `while` pour demander un nombre entier positif :

- Utilisation d'une boucle `while True` pour continuer à demander un nombre jusqu'à ce que l'utilisateur entre un entier positif.
- Utilisation d'un bloc `try-except` pour attraper les erreurs si l'utilisateur n'entre pas un nombre entier (`ValueError`).
- Vérification si le nombre est positif. Si oui, utilisation de `break` pour sortir de la boucle. Sinon, affichage d'un message d'erreur et la boucle continue.

3. **Affichage de la table de multiplication :**

- Utilisation de la fonction `print` pour afficher un en-tête pour la table de multiplication.
- Utilisation d'une boucle `for` avec `range(1, 11)` pour générer les lignes de la table de multiplication.
- Calcul et affichage de chaque ligne de la table avec une multiplication simple.

4. **Demande à l'utilisateur s'il veut recommencer ou quitter :**

- Utilisation de la fonction `input` pour demander à l'utilisateur s'il veut entrer un autre nombre ou quitter.
- Conversion de la réponse en minuscules et suppression des espaces superflus avec `strip().lower()` .
- Vérification de la réponse. Si elle n'est pas 'oui', affichage d'un message d'au revoir et utilisation de `break` pour sortir de la boucle principale.
- Si la réponse est 'oui', utilisation de `continue` pour recommencer la boucle principale.