

---

# JEU D' ESSAI COMPLET – PROJET CAMPING LA CERISAIE

---

## Étapes préalables (Avant de lancer le programme)

### ◇ 1. Organisation des fichiers

Vérifiez que ces fichiers sont dans le même dossier :

typeEmplacement.h

typeEmplacement.c

emplacement.h

emplacement.c

main.c

---

### ◇ 2. Compilation

Si vous utilisez un terminal :

```
gcc main.c typeEmplacement.c emplacement.c -o camping
```

---

### ◇ 3. Nettoyage initial

Supprimer les fichiers suivants s' ils existent :

types.dat

emplacements.dat

Cela permet de repartir sur une base propre.

---

## ❏ Scénario du jeu d'essai

---

### ◇ Étape A : Premier lancement (Base vide)

Action :

Lancer le programme : ./camping

Résultat attendu : Info : Fichier 'types.dat' non trouve

Vérification :

- Choisir option 1 (Voir les tarifs)
- La liste doit être vide

---

### ◇ Étape B : Création des types

Ajouter les types suivants (via ton module ou insertion temporaire) :

Code	Nom	Prix
1	Caravane	13.50
2	Tente	11.00
3	CampingCar	14.00

Code	Nom	Prix
4	Bungalow	17.50

Quitter avec option 0 pour sauvegarder.

---

### ◇ Étape C : Test de persistance (IMPORTANT)

1. Lancer le programme
2. Ajouter les types
3. Quitter avec option 0
4. Vérifier qu' un fichier types.dat apparaît
5. Relancer le programme
6. Choisir option 1

Résultat attendu :

Les types doivent apparaître automatiquement.

✓ Cela prouve que le chargement fonctionne.

---

## 3 Tests du module EMPLACEMENT

---

### ◇ Étape D : Ajout d' emplacements

Ajouter les emplacements suivants :

Num Surface Capacité Type

Num Surface Capacité Type

1	80	4	2
2	120	6	4
3	90	5	1
4	60	2	2

Résultat attendu : Emplacement ajoute avec succes.

Puis option 5 (Afficher tous les emplacements)

Les emplacements doivent apparaître avec état = Libre.

---

#### ◇ **Étape E : Test d' attribution**

Choisir option 6 :

- Code type : 2 (Tente)
- Nombre personnes : 3

Résultat attendu :

Un emplacement libre de type 2 doit être attribué.

Son état devient : Occupé.

Vérifier avec option 5.

---

### ◇ **Étape F : Test de libération**

Choisir option 7 :

- Numéro emplacement attribué

Résultat attendu :

Etat devient Libre.

---

### ◇ **Étape G : Test liste par type**

Option 8 :

Entrer code 4 (Bungalow)

Résultat attendu :

Seuls les bungalows doivent apparaître.

---

### ◇ **Étape H : Recherche technique**

Option 9 :

Surface minimale : 70

Capacité minimale : 4

Résultat attendu :

Seuls les emplacements respectant ces critères apparaissent.

---

## **4. Tests des cas limites**

---

◇ **Test 1 : Dépassement MAX\_TYPES**

Essayer d' ajouter 11 types.

Résultat attendu :

Le programme ne plante pas.

La boucle while (nbTypes < MAX\_TYPES) empêche le dépassement.

---

◇ **Test 2 : Type inexistant**

Ajouter un emplacement avec : Code type : 99

Résultat attendu : Type inexistant.

---

◇ **Test 3 : Attribution impossible**

Essayer d' attribuer :

- Type : 4
- Nombre personnes : 20

Résultat attendu : Aucun emplacement disponible.

---

◇ **Test 4 : Suppression logique**

Supprimer emplacement 2.

Puis afficher la liste.

Résultat attendu :

L' emplacement supprimé ne doit plus apparaître.

---

## **5 Vérification des fichiers générés**

---

Après les tests :

Le dossier doit contenir :

types.dat

emplacements.dat

Cela prouve que la persistance fonctionne.

---

## **🎯 Conclusion du jeu d' essai**

Le programme est validé si :

- ✓ Chargement au démarrage fonctionne
- ✓ Sauvegarde à la fermeture fonctionne
- ✓ Ajout / Modification / Suppression fonctionnent
- ✓ Attribution respecte disponibilité
- ✓ Recherche fonctionne

- ✓ Aucun plantage sur cas limite