

TD 2 - C, Javascript - Types des données

Hugo Bolloré¹

¹hugo.bollore@uvsq.fr

Exercice 2.1 Sûreté du typage en C

Question 2.1.1 Conversions implicites et explicites

Dans le code suivant, quelles sont les conversions implicites et les conversions explicites ?

```
int main() {  
    int i = 0;  
    void *v = &i;  
    char *c = v;  
    unsigned int *u = (unsigned int *)c;  
    return *u;  
}
```

Copiez, compilez et exécutez ce programme. Comment qualifieriez vous la sûreté du typage en C dans cet exemple ?

Question 2.1.2 Intégrité du type des variables

Copiez et compilez le code suivant :

```
#include <stdio.h>  
#include <limits.h>  
  
void main() {  
    int maValeur = INT_MAX;  
    unsigned int* overflow = &maValeur;  
    (*overflow)++;  
  
    printf("%d\n", maValeur);  
}
```

Expliquez le résultat. Comment qualifieriez vous la sûreté du typage en C dans cet exemple ?

Exercice 2.2 Sûreté du typage en Javascript

Nous allons utiliser le langage Javascript par la suite, c'est un langage objet dont la syntaxe utilisé dans ce TD est proche de ce que vous avez déjà utilisé.

Dans votre navigateur, allez sur le site <https://playcode.io/javascript>

Vous pouvez également télécharger le package *nodejs*, écrire le code dans un fichier `script_name.js` et le lancer avec :

```
node script_name.js
```

Question 2.2.1 Compatibilité et équivalence des types

Exécutez le code Javascript suivant :

```
class Poule {
  oeufsPondus = 42;
  CotCot() {
    console.log("444719")
  }
}

class Perroquet {
  nom = "Robert";
  CotCot() {
    console.log("Cot Cot Cot C'est un oeuf !")
  }
}

function Pondus (objet) {
  objet.CotCot();
}

let poule = new Poule();
let perroquet = new Perroquet();

Pondus(poule);
Pondus(perroquet)
```

D'après le résultat de ce code, quel système est utilisé pour **comparer** les types en Javascript ?

Question 2.2.2 Vérification du typage

Exécutez le code suivant

```
studentId = 42;
studentNumber = '42';

console.log(typeof studentId); // Affiche le type de la variable studentId
console.log(typeof studentNumber); // Affiche le type de la variable studentNumber

studentId = '42';
studentNumber = 42;

console.log(typeof studentId); // Affiche le type de la variable studentId
console.log(typeof studentNumber); // Affiche le type de la variable studentNumber
```

Comment qualifieriez vous le système de vérification du type en Javascript ?

Exercice 2.3 Conversions**Question 2.3.1** Conversions implicites en Javascript

Devinez et testez le résultat des opérations suivantes :

- "1" + 1
- 1 + "1"
- "2" - 1
- 3 < 2 < 1

Pour chacune de ces opérations, expliquez pourquoi le langage renvoie ces valeurs.

Question 2.3.2 Conversions implicites en C

```
#include <stdio.h>

void main()
{
    int i_value    = 16777217;
    float f_value = 16777216.0;
    printf("The integer is: %d\n", i_value);
    printf("The float is:    %f\n", f_value);
    printf("Are they equal ? %s\n", i_value == f_value ? "true" : "false" );
}
```

Expliquez les résultats de ce programme.

Cet exemple s'inspire de l'article Wikipedia sur les conversions de type [1]

Question 2.3.3 Conversions et qualificateurs de types

Considérez le programme suivant

```
void main()
{
    int i = 42;

    int const * ptr_ic = 0;
    ptr_ic = &i;
    i = *ptr_ic + 1;
    *ptr_ic += 1;
    *(int *)ptr_ic += 1;
}
```

Excepté la déclaration de `i`, expliquez exactement ce que font toutes les lignes et pourquoi l'avant dernière ligne n'est pas acceptée par le compilateur alors que la précédente et la suivante le sont.

Exercice 2.4 Cast

En utilisant le fichier *data.set* qui vous sera fourni, calculez la différence entre les 50 valeurs des tableaux *uintVals* et *intVals*, stockez celles-ci dans des *long int*, et calculez et affichez la moyenne de toutes les différences qui sont inférieures à 777.

Pour pouvoir lire les données binaires contenues dans le fichier *data.set* vous devrez partir du code suivant (le fichier *data.set* devra être copié dans le même répertoire que votre programme compilé) :

```
#include <stdio.h>
#include <stdlib.h>

#define NB_VALS 50

void main()
{
    FILE* fp;
    unsigned int uintVals[NB_VALS];
    int intVals[NB_VALS];
    int returnValue = 0;

    fp = fopen("data.set", "rb");
    if (fp == NULL) {
        fprintf(stderr, "Cannot open file data.set\n");
        exit(-1);
    }

    returnValue = fread(uintVals, sizeof(unsigned int), NB_VALS, fp);
    if (returnValue != NB_VALS) {
        fprintf(stderr, "Cannot read %d blocks in file data.set\n", NB_VALS);
        exit(-1);
    }

    returnValue = fread(intVals, sizeof(int), NB_VALS, fp);
    if (returnValue != NB_VALS) {
        fprintf(stderr, "Cannot read %d blocks in file data.set\n", NB_VALS);
        exit(-1);
    }
}
```

Vous devriez obtenir une moyenne de -1306717103.666667.

References

- [1] Wikipedia contributors. Type conversion — Wikipedia, the free encyclopedia, 2024. [Online; accessed 23-November-2024].