TP5_INF203

Groupe: INF2

Binôme: Shaghayegh HAJMOHAMMADKASHI

Kaiwen ZHENG

Terminal : cd INF203

./TP1/scripts/installeTP.sh 5

cd TP5

[a]:

```c
#include <stdio.h>
#include "generer_entier.c"

// int main() {
//     long x = generer_entier(100);
//     printf("J'ai genere l'entier %ld\n", x);
//     return 0;
// }

// int main() {
//     long x = generer_entier(100);
//     if (x > 42) {
//      printf("Trop grand\n");
//     } else if (x < 42) {
//      printf("Trop petit\n");
//     } else {
//      printf("Youpi\n");
//     }
//     return 0;
// }

// avec une boucle

int main() {
    int n;
    printf("donner le nombre d'entiers à tester : ");
    scanf("%d", &n); // lire la valeur entrée par l'utilisateur // scanf :
nous avons stocké cette valeur dans la variable n à l'aide de cette ft
    // i++ pour dire i=i+1
    for (int i = 0; i < n; i++) {
        long x = generer_entier(100);
        if (x > 42) {
            printf("Trop grand\n");
        } else if (x < 42) {
```

```
            printf("Trop petit\n");
        } else {
            printf("Youpi\n");
        }
    }
    return 0;
}
```

[b] :

La taille d'une variable de type unsigned char est de 1 octet, soit 8 bits. Le plus grand entier représentable avec ce type est 2^8 - 1, soit 255.

[c] :

deborde_short.c

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

int main() {
    unsigned short int courant = 0, prochain = 1;

    printf("taille du type unsigned short int : %ld octet(s)\n",
sizeof(unsigned short int));

    while (prochain > courant) {
        courant = prochain;
        prochain = prochain + 1;
    }

    printf("%u + 1 = %u donc ...\n", courant, prochain);
    printf("valeur maximum d'une variable de type unsigned short int : %u\n",
courant);

    return 0;
}
```

La taille d'une variable de type unsigned short int est de 2 octets, soit 16 bits. Le plus grand entier représentable avec ce type est 2^16 - 1, soit 65535.

[d] :

deborde_int.c

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

int main() {
```

```
    unsigned int courant = 0, prochain = 1;

    printf("taille du type unsigned int : %ld octet(s)\n", sizeof(unsigned
int));

    while (prochain > courant) {
        courant = prochain;
        prochain = prochain + 1;
    }

    printf("%u + 1 = %u donc ...\n", courant, prochain);
    printf("valeur maximum d'une variable de type unsigned int : %u\n",
courant);

    return 0;
}
```

La taille d'une variable de type ‘unsigned int’ est de 4 octets, soit 32 bits. Le plus grand entier représentable avec ce type est 2^32 - 1, soit 4294967295.

[e] :

hajmohas@im2ag-turing-01:[~/INF203/TP5]: gcc -o deborde_char deborde_char.c

hajmohas@im2ag-turing-01:[~/INF203/TP5]: ./deborde_char

taille du type unsigned char : 1 octet(s)

255 + 1 = 0 donc ...

valeur maximum d'une variable de type unsigned char : 255

hajmohas@im2ag-turing-01:[~/INF203/TP5]: gcc -o deborde_short deborde_short.c

hajmohas@im2ag-turing-01:[~/INF203/TP5]: ./deborde_short

taille du type unsigned short int : 2 octet(s)

65535 + 1 = 0 donc ...

valeur maximum d'une variable de type unsigned short int : 65535

hajmohas@im2ag-turing-01:[~/INF203/TP5]: gcc -o deborde_int deborde_int.c

hajmohas@im2ag-turing-01:[~/INF203/TP5]: ./deborde_int

taille du type unsigned int : 4 octet(s)

4294967295 + 1 = 0 donc ...

valeur maximum d'une variable de type unsigned int : 4294967295

hajmohas@im2ag-turing-01:[~/INF203/TP5]: time ./deborde_int

taille du type unsigned int : 4 octet(s)

4294967295 + 1 = 0 donc ...

valeur maximum d'une variable de type unsigned int : 4294967295

real 0m8,723s

user 0m8,712s

sys 0m0,011s

hajmohas@im2ag-turing-01:[~/INF203/TP5]: time ./deborde_short

taille du type unsigned short int : 2 octet(s)

65535 + 1 = 0 donc ...

valeur maximum d'une variable de type unsigned short int : 65535

real 0m0,018s

user 0m0,016s

sys 0m0,003s

Le temps d'exécution de deborde_int est d'environ 9 secondes, tandis que celui de deborde_short est d'environ 0,02 secondes. Cela signifie que le calcul pour deborde_int prend beaucoup plus de temps que celui pour deborde_short. Cela est dû au fait que unsigned int occupe plus d'espace mémoire que unsigned short int. Lorsque nous exécutons la boucle while, il y a beaucoup plus de valeurs à traiter pour unsigned int que pour unsigned short int, ce qui entraîne une durée d'exécution beaucoup plus longue.

[f] :

deborde_long:

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

int main() {
    unsigned long long courant = 0, prochain = 1;

    printf("taille du type unsigned long long : %ld octet(s)\n",
sizeof(unsigned long long));

    while (prochain > courant) {
        courant = prochain;
        prochain = prochain + 1;
    }

    printf("%u + 1 = %u donc ...\n", courant, prochain);
    printf("valeur maximum d'une variable de type unsigned long long : %u\n",
courant);
```

```
    return 0;
}
```

hajmohas@im2ag-turing-01:[~/INF203/TP5]: gcc -o deborde_long deborde_long.c

hajmohas@im2ag-turing-01:[~/INF203/TP5]: time ./deborde_long

taille du type unsigned long long : 8 octet(s)

Pensez-vous devoir essayer pour avoir la réponse ?Non

[g] :

```c
#include <stdio.h>
#include "generer_entier.c"

void echanger(long Tab[], int i, int j) {
    long tmp;
    tmp = Tab[i];
    Tab[i] = Tab[j];
    Tab[j] = tmp;
}

/* inserer a sa place l'entier val dans la sequence triee Tab[0..nb-1] */
void inserer(long Tab[], int nb, int val) {
    Tab[nb] = val; // Place l'élément val en position nb
    int i = nb;
    while (i > 0 && Tab[i] < Tab[i-1]) { // Échange val avec son voisin de
gauche tant qu'il n'est pas à sa place
        echanger(Tab, i, i-1);
        i--;
    }
}

// affiche a l'écran T[0..nb-1]
void afficher(long T[], int nb) {
    int i;
    printf("[ ");
    for (i = 0; i < nb; i++) {
        printf("%ld ", T[i]);
    }
    printf("]\n");
}

int main() {
    int Taille = 20;
    long T[Taille];
    int i;
```

```
    long valeur ;

    for (i = 0; i < Taille; i++) {
        valeur = generer_entier(100) ;
        inserer(T, i, valeur); // Insère la valeur dans le tableau et le trie
        afficher(T, i+1); // Affiche le tableau jusqu'à la ième case remplie
    }

    return 0;
}
```

La fonction inserer est appelée dans la boucle for de la fonction main avec les paramètres effectifs suivants :

Tab: le tableau T

nb: la variable i, qui représente le nombre d'éléments déjà présents dans le tableau (c-à-d l'index de la prochaine case vide)

val: la variable valeur, qui contient la valeur générée aléatoirement par la fonction generer_entier(100) à chaque itération de la boucle.

[h] :

La commande tail -n 4 affiche les 4 dernières lignes du fichier, tandis que la commande tail -n +4 affiche toutes les lignes à partir de la quatrième ligne du fichier.

- Par exemple, si on a un fichier mon_fichier.txt contenant les lignes suivantes :

Première ligne

Deuxième ligne

Troisième ligne

Quatrième ligne

Cinquième ligne

- La commande tail -n 4 mon_fichier.txt affichera :

Deuxième ligne

Troisième ligne

Quatrième ligne

Cinquième ligne

- Tandis que la commande tail -n +4 mon_fichier.txt affichera :

Quatrième ligne

Cinquième ligne

[i] :

```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 fichier"
    exit 1
fi

if [ ! -f $1 ]; then
    echo "Erreur: $1 n'est pas un fichier"
    exit 2
fi

count=0

while read line; do
    echo "$line"
    ((count++))
    if [ $count -eq 4 ]; then
        echo "...."
        count=0
    fi
done < $1

if [ $count -ne 0 ]; then
    echo "...."
fi
```