uOttawa

CSI 3520 - Concepts des langages de programmation

Automne 2022 Section A

Professeur Kalonji Kalala

Devoir 3

Kien Do (ID: 300163370)

1. **Réponse**

```r
# Create a vector `names` that contains your name and the names of 2 people
# next to you. Print the vector.
# Use the colon operator : to create a vector `n` of numbers from 10:49
# Use the `length()` function to get the number of elements in `n`
# Add 1 to each element in `n` and print the result
# Create a vector `m` that contains the numbers 10 to 1 (in that order).
# Hint: use the `seq()` function
# Subtract `m` FROM `n`. Note the recycling!
# Use the `seq()` function to produce a range of numbers from -5 to 10 in
↪  `0.1`
# increments. Store it in a variable `x_range`

names <- c("Kien", "Evan", "Shivan");
print(names);

n <- c(10:49);
print(length(n));
n <- n + 1;
print(n);


m <- seq.int(10, 1); # peut être c(10:1) également
print(m)

print(n - m);

x_range <- seq(-5, 10, 0.1);
print(x_range);
```

2. **Réponse**

Code en C

```c
#include <stdio.h>

int main()
{
    int j = 0;
    int k = (j + 13) / 27;

    while (k <= 10)
    {
        k = k + 1;
        int i = 3 * k - 1;
    }

    return 0;
}
```

Code en Java

```java
public class Main
{
    public static void main(String[] args) {
        int j = 0;
        int k = (j + 13) / 27;

        while (k <= 10) {
            k = k + 1;
            int i = 3 * k - 1;
        }
    }
}
```

Code en Python

```python
j = 0
k = (j + 13) / 27

while k <= 10:
    k = k + 1
    i = 3 * k - 1
```

Meilleure écriture: Python
— Très facile et court à écrire. Pas besoin de spécifier les types des données.
Meilleure lisibilité: C
— Très facile à lire. Les accolades de chaque bloque de code commencent et terminent sur la même ligne verticale. Les types des données sont spécifiés, ce qui améliore la lisibilité.
Meilleure combinaison: Java
— Plus facile à lire par rapport à Python et plus facile à écrire par rapport à C.

3. **Réponse**

Les valeurs de sum1 et sum2 si les opérandes dans les expressions sont évalués de:

(a) Gauche à droite

sum1 = 46
sum2 = 48

(b) Droite à gauche

sum1 = 48
sum2 = 46

4. **Réponse**

Le programme de l'exercice 3 en C++

```cpp
#include <iostream>

using namespace std;

int fun(int *k)
{
    *k += 4;
    return 3 * (*k) - 1;
}

int main()
{
    int i = 10, j = 10, sum1, sum2;
    sum1 = (i / 2) + fun(&i);
    sum2 = fun(&j) + (j / 2);

    return 0;
}

// Résultats:
// sum1 = 46 et sum2 = 48
```

Le programme de l'exercice 3 en Java

```java
public class Main
{
    public static int fun(int k) {
        k += 4;
        return 3 * k - 1;
    }

    public static void main(String[] args) {
        int i = 10, j = 10, sum1, sum2;
        sum1 = (i / 2) + fun(i);
        sum2 = fun(j) + (j / 2);
    }
}

// Résultats:
// sum1 = 46 et sum2 = 46
```

— Les deux versions sont différentes parce que en Java, les entiers sont passés par valeur, et non passés par addresse, et vice versa. C'est-à-dire, en C++, les entiers i et j changent en main() car ils étaient changés en fun(). Ça fonctionne parce que ces deux entiers étaient passés par référence. Par contre, en Java, les entiers i et j ne changent pas en main() même s'ils étaient changés en fun(), parce que ces entiers sont passés par valeur.

5. **Réponse**

Écrivez un programme en Java, C++, Python, qui effectue un grand nombre d'opérations en virgule flottante et un nombre égal d'opérations sur des nombres entiers et comparez le temps requis.

Java

```java
public class Main {
    public static void main(String[] args) {
        float total = 0;
        int count = 100000000;
        for (int i = 0; i < count; i++) {
            total += 0.1;
        }
    }
}
```

C++

```cpp
#include <iostream>

using namespace std;

int main()
{
    float total = 0.0;
    int count = 100000000;
    for (int i = 0; i < count; i++)
    {
        total += 0.1;
    }

    return 0;
}
```

Python

```python
total = 0.0
count = 100000000
for x in range(count):
    total += 0.1
```

Java: ∼3 secondes
— Vitesse moyenne car Java est compilé et interprété.
C++: ∼1 seconde
— Vitesse la plus rapide car C++ est compilé et est très bon pour les calculs mathématiques.
Python: ∼9 secondes
— Vitesse la plus lente car Python est interprété.