

4.2.4 Tabla de Permutación

4.2.4.1 Implementación en lenguaje C.

Deberán usar las funciones `setSeed` y `nextChar` tal como se ven a continuación.

La función `setSeed` deberá tomar la semilla elegida y la función `nextChar` permitirá obtener un valor en el rango [0,255]

Estas funciones son equivalentes a las que usa Java. Es importante tener en cuenta que la función `setSeed` trabaja sobre un valor de 48 bits, por ese motivo se usa un tipo adecuado en C, que sería equivalente a un `long long int`.

```
void
setSeed(int64_t s)
{
    seed = (s ^ 0x5DEECE66DL) & ((1LL << 48) - 1);
}
uint8_t
nextChar(void){
    seed = (seed * 0x5DEECE66DL + 0xBL) & ((1LL << 48) - 1);
    return (uint8_t)(seed>>40);
}
```

A continuación, un ejemplo completo. La variable `seed` es global.

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

#define MAX 50
#define SET 10
/*variable global*/
int64_t seed; /*seed debe ser de 48 bits; se elige este tipo de 64 bits*/

void setSeed(int64_t seed);
uint8_t nextChar(void); /*devuelve un unsigned char*/

int
main()
{
    int i;
    uint8_t num;
    setSeed(SET);
    for (i = 0; i < MAX; i++)
    {
        num = nextChar();
        printf("%d\t", num);
    }
    return EXIT_SUCCESS;
}

void
setSeed(int64_t s)
{
    seed = (s ^ 0x5DEECE66DL) & ((1LL << 48) - 1);
}
uint8_t
nextChar(void){
    seed = (seed * 0x5DEECE66DL + 0xBL) & ((1LL << 48) - 1);
    return (uint8_t)(seed>>40);
}
```

4.2.4.1 Implementación en lenguaje Java.

Se usará la implementación de `java.util.Random`, que es equivalente a la implementación anterior en C.

Se muestra un ejemplo de cómo usarla:

```
import java.util.Random;
public class pruebaRandom {

    public static void main(String[] args) {
        Random rnd = new Random();
        rnd.setSeed(10);
        for(int i=0; i < 50; i++)
        {
            int num = rnd.nextInt(256); /*genera nro en [0,255]*/
            System.out.println(num);
        }
    }
}
```