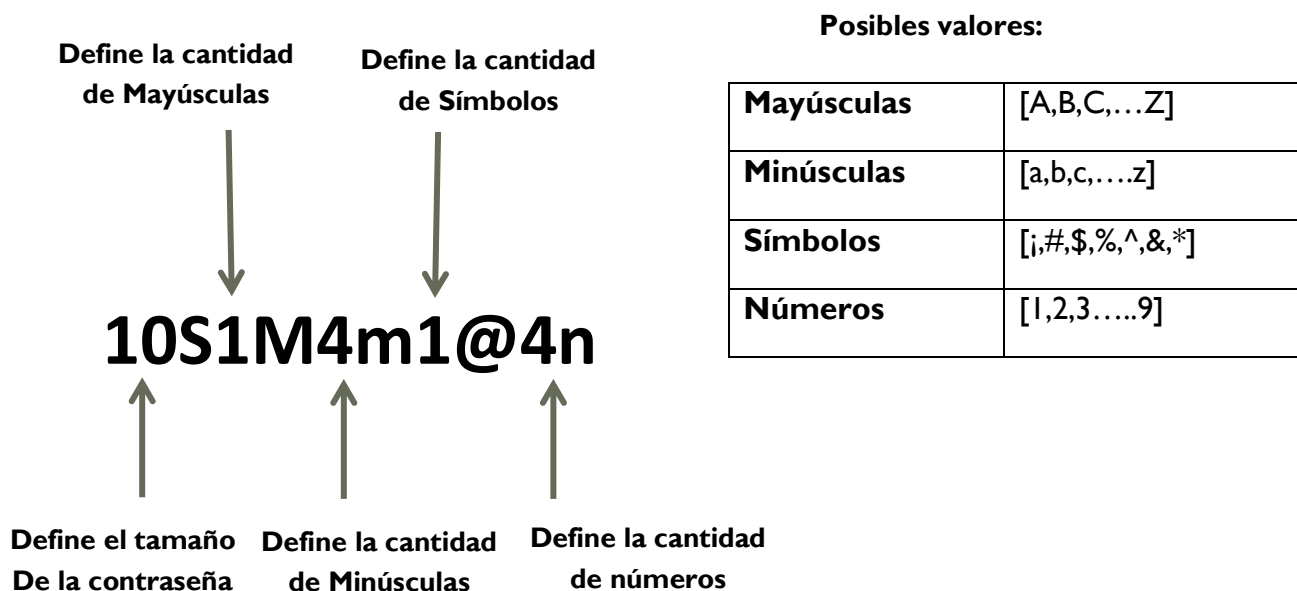


Competencia

Aplicar los conocimientos básicos de funciones simples, que regresan valor y paramétricas a través del desarrollo de una aplicación que involucre el uso de estos conceptos.

Ejercicio I

Realizar un programa en C que permita generar contraseñas, para generar éstas se deberán utilizar patrones con la siguiente estructura:



Este ejemplo generaría un resultado como:

Np\$5g7u2r9

Deberá considerarse lo siguiente:

1. Cada carácter (Mayúsculas, minúsculas, símbolos y números) serán elegidos de manera aleatoria y la generación de la contraseña será un carácter de cada tipo a la vez, es decir, éstas serán colocadas una por una hasta que se hayan acabado. Durante la generación de la contraseña los caracteres se irán intercalando en el orden mayúsculas, minúsculas, símbolos y números hasta que se agoten. En caso de que un tipo de carácter agote su número de repeticiones, no aparecerá y se continuara con el siguiente.

2. El patrón deberá contener obligatoriamente los identificadores:
 - **S:** para el tamaño de la contraseña
 - **M:** para las Mayúsculas
 - **m:** para las minúsculas.
 - **@:** para los símbolos
 - **n:** para los números.
3. Se deberá validar que exista el valor numérico antes de cada identificador, pues de no ser así la contraseña no se puede generar.
4. Validar que el tamaño de la contraseña sea mayor o igual a la suma de la cantidad especificada de cada una de las características.

Por ejemplo, en la figura donde se muestra la estructura tenemos: 10SIM4MI@4n donde se cumple que $10 \geq 1 + 4 + 1 + 4$, pero en el caso que tuviéramos un número mayor a 10, la contraseña no puede ser generada, esto se debe notificar al usuario. **En caso** de que la suma de los números de cada identificador sea **menor** que el tamaño requerido, se rellenara con un carácter aleatorio.

5. El programa deberá contener un menú con las siguientes opciones:
 - (a) **Almacenar patrón:** Esta opción permitirá registrar nuevos patrones de caracteres y almacenarlos dentro de un arreglo, sin exceder su capacidad (20 patrones)
 - (b) **Listar patrones registrados:** Esta opción permitirá listar todos los patrones registrados, así como las contraseñas que éstos generaron.
 - (c) **Eliminar patrón:** Esta opción permitirá eliminar un patrón del arreglo, para eso la función recibirá el arreglo de patrones y la cantidad de patrones que tiene. Esta función no regresará ningún valor.
 - (d) **Generar Contraseña:** Esta opción generará la contraseña de un patrón al azar de los que se encuentran almacenados en el arreglo de patrones (Para esto se generará un número aleatorio entre 1 y la cantidad de patrones almacenados) y se mostrará el resultado que el patrón generó. Se deberá validar que la contraseña mostrada no sea igual a ninguna de las últimas 4 generadas. Se deberá tener una función que genere el número aleatorio y que reciba un arreglo que contendrá únicamente los últimos números generados para verificar que el número nuevo no esté en los últimos cuatro.

Ejercicio 2

Desarrolle un programa en C que reciba una o dos cadenas que serán ingresadas al momento de la ejecución del programa desde la línea de comandos y ofrecerá un menú con las siguientes opciones.

- **Comparar cadenas:**

Esta opción utilizará una función con el prototipo **int Comparar(char[], char[])** que regresará un 0 si las cadenas son iguales, si la primera cadena es más grande regresará un 1, si la segunda es más grande, regresará un 2.

- **Comprobar palíndromo:**

Esta opción utilizará una función con el prototipo **int comprobarPalindromo(char[])** y regresará un 1 si la cadena es un palíndromo, es decir, si la cadena ingresada se lee de igual manera de izquierda a derecha y de derecha a izquierda, regresará un -1 en el caso contrario.

- **Mayúsculas-Minúsculas:**

Esta opción tendrá el prototipo **int mayusculasMinusculas(char [])** y permitirá cambiar los caracteres en mayúscula de la cadena, a minúscula y viceversa. Si la operación se realizó sin problemas regresará un 0, de lo contrario regresará un 1.