

Reporte de laboratorio 1

Laura Rincón Riveros - B55863
Esteban Vargas Vargas - B16998

4 de septiembre de 2016

Índice

1. Introducción	1
2. Código en C++	2
3. Conclusiones	5

1. Introducción

En el presente laboratorio se realizó un programa capaz de traducir una secuencia de ARN en su respectiva cadena de aminoácido, como se presenta a continuación:

2. Código en C++

```
string variables [64] = {"UUU","UUC","UUA","UUG","CUU","CUC","CUA","CUG","AUU",
"AUC","AUA","AUG","GUU","GUC","GUA","GUG","UCU","UCC","UCA","UCG","CCU","CCC",
"CCA","CCG","ACU","ACC","ACA","ACG","GCU","GCC","GCA","GCG","UAU","UAC","UAA",
"UAG","CAU","CAC","CAA","CAG","AAU","AAC","AAA","AAG","GAU","GAC","GAA","GAG",
"UGU","UGC","UGA","UGG","CGU","CGC","CGA","CGG","AGU","AGC","AGA","AGG","GGU",
"GGC","GGA","GGG"};
char equival [64] = {'F','G','L','L','L','L','L','L','L','I','I','I','M','V','V','V',
'V','S','S','S','S','S','P','P','P','P','T','T','T','T','A','A','A','A','Y','Y','|',
'|','H','H','Q','Q','N','N','K','K','D','D','E','E','C','C','|','W','R','R','R',
'R','S','S','R','R','G','G','G','G'};
```

Figura 1: Variables globales del programa

En la figura 1 se presentan dos arreglos globales; *variables* que contiene las combinaciones de bases nitrogenadas (ARN) y el arreglo *equival* que contiene la correspondiente traducción a aminoácidos representados por siglas de una letra mayúscula. Estos dos arreglos se trabajan como una especie de diccionario.

```
/**
 * @brief Cuenta el numero de caracteres recibidos por consola.
 *
 *
 */
int contarchars(char* arg)
{
    int cont=0;
    char h;
    do{
        h=arg[cont];
        cont++;
    }
    while(h!='\0');

    int contador=cont-1;
    return contador;
}
```

Figura 2: Función contarchars

La función contar (figura 2) fue necesaria para lograr saber el tamaño de la cadena de ARN ingresada. Esta información fue de mucha importancia para lograr la tarea a realizar.

```

/**
 * @brief Traduce el ARN en una cadena de aminoacidos.
 *
 */
char* traducirARNaAA(char* arn)
{
    int m= contarchars(arn);
    int i=0;
    int j=0;
    int x=0;
    char* codones = new char [m/3];

    for(i=0;i<m;i+=3){
        char* cadena = new char[3];
        cadena[0] = arn[i];
        cadena[1]= arn[i+1];
        cadena[2]= arn[i+2];

        for (j=0;j<64; j++) {
            if (cadena==variables[j]){
                codones[x]= equival[j];
                x++;
            }
        }
        delete [] cadena;
    }
    return codones;
}

```

Figura 3: Función traducirARNaAA

En la figura 3 se presenta el código de la función traducirARNaAA, que recibe un puntero de char el cual compara con el arreglo *variables* y reemplaza con su respectiva sigla del arreglo *equival*. Para ello se utilizó memoria dinámica en donde el puntero *cadena*, guarda una sección del ARN, compara, traduce, libera la memoria del mismo y reinicia el ciclo hasta traducir todo el ARN para finalmente devolver un arreglo de char con los aminoácidos.

```

/**
 * @brief Imprime un arreglo de char.
 *
 *
 */
void imprimirArregloDeChar(char* imp, int n)
{
    int i=0;
    for(i=0;i<n;i++){
        cout<<imp[i];
    }
    cout<<endl;
}

int main (int argc, char** argv)
{
    imprimirArregloDeChar(traducirARNaAA(argv[1]),contarchars(argv[1]));

    return 0;
}

```

Figura 4: Función `imprimirArregloDeChar` y `main`

En la figura 4 se presenta la función `imprimirArregloDeChar` que recorre el arreglo según su tamaño e imprime cada elemento.

Finalmente, en la figura se muestra también el método `main`, el cual llama a la función `imprimirArregloDeChar` la cual necesita dos parámetros como se puede observar:

- Un puntero char a imprimir que se obtiene del puntero char que devuelve la función `traducirARNaAA`, véase la figura 3.
- Un entero tamaño de la cadena a imprimir que se obtiene de la función `contarchars`, véase la figura 2.

Y se obtiene:

```

laura@macondo:~/Documents/Estructuras/laboratorios/lab1$ make
g++ -Wall -o out.o lab1.cpp
echo "programa en C++"
programa en C++
./out.o UAACCUUCUACUACGUAG
|PSTT|
rm out.o
laura@macondo:~/Documents/Estructuras/laboratorios/lab1$ █

```

Figura 5: Resultado del programa en consola

3. Conclusiones

- Fue necesario el uso de memoria dinámica para resolver el problema planteado.
- Se logró traducir una cadena de codones ARN a los aminoácidos respectivos mediante la elaboración de un programa en C++.