



DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE CONCEPCIÓN
CONCEPCIÓN, CHILE.

Informe N°1

Calculando factoriales

Profesor: Mario Medina

Alumno: Aldo Mellado Opazo

24 de agosto de 2018

Programación Orientada al Objeto
Ingeniería Civil en Telecomunicaciones

1. Calculando Factoriales

Para este problema se debía calcular.^{el} factorial de 1000 y sumar sus dígitos. Para ello, se debía tener en consideración que las variables de tipo `int`, `long int`, `long long int`, no dan a basto para almacenar valores de factorial tan grandes -notar que $1000!$ es $\approx 4,0238726007 \cdot 10^{2567}$ - de modo que debía buscarse una solución alternativa.

Para estos efectos se consideró trozar cada número obtenido en el cálculo del factorial de la siguiente manera.

$$10! = \begin{array}{|c|c|c|c|c|c|c|} \hline 3 & 6 & 2 & 8 & 8 & 0 & 0 \\ \hline \end{array}$$

Tabla 1: Representación de vector resultante

sin embargo, el realizar la multiplicación término a término, de un vector por otro, aplicando a la suma un carry y al producto, la utilización de esta función suponía una operación demasiado engorrosa y que en la iteración anterior de esta tarea no dió resultado, razón por la cual se procedió de otra manera.

1.1. Problemas

Más que aparecer como problemas aparecen como alcances o salvedades que se hicieron, y es que dado que el factorial no está definido para los números negativos, se le dió la posibilidad de que el usuario si es que desee calcular un valor menor que cero, pudiera reingresar el valor, a posteriori, de hacerle notar que el valor ingresado no es válido.

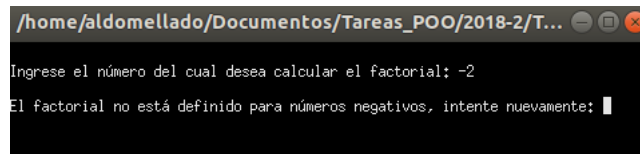


Fig. 1: Valor ingresado no válido

1.1.1. Solución

Tal como se mencionó anteriormente, para el cálculo de los factoriales, hacerlo por una lista resultó engorroso y complejo de manera que se hizo tal que para un vector, se fuera calculando el resultado de hasta el número elegido por el usuario, donde dicho número era descompuesto dígito a dígito.

Una vez descompuesto, este era multiplicado por el siguiente número dentro del factorial. Por ejemplo, para el caso de $11!$;

$$\begin{aligned}
 10! &= \begin{array}{|c|c|c|c|c|c|c|} \hline 3 & 6 & 2 & 8 & 8 & 0 & 0 \\ \hline \end{array} \\
 11! &= 10! * 11 \\
 &= \begin{array}{|c|c|c|c|c|c|c|} \hline 3 & 6 & 2 & 8 & 8 & 0 & 0 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 1 & 1 \\ \hline \end{array} \\
 &= \begin{array}{|c|c|c|c|c|c|c|} \hline 3 & 9 & 9 & 1 & 6 & 8 & 0 & 0 \\ \hline \end{array}
 \end{aligned}$$

Ello se hace a través de dos ciclos for, como se enuncia en el siguiente fragmento:

```

1  for(int i=1;i<=num;i++)
2  {
3      for(int j=0;j<auxiliar.size();j++)
4      {
5          int temp = auxiliar[j]*i; // x contiene el producto digito a digito
6          auxiliar[j]=(temp+c)%10; //Contiene el digito almacenado en la posicion j-esima
7          c = (temp+c)/10; //Contiene el valor del acarreo que será; almacenado mas tarde en los indices
8      }
9      while(c>0) //el ciclo while almacena el valor de acarreo en el vector
10     {
11         auxiliar.push_back(c%10);
12         c = c/10;
13     }
14 }

```

En adición a lo anterior, la tarea solicita que se cree una función que una vez calculado el factorial, realice la suma de los dígitos, sin embargo, para efectos de como opera el código, se tiene que el cálculo de los dígitos se va haciendo en reversa, es decir, la forma en que se almacena el factorial de 10! en `vector<int> auxiliar` es la siguiente:

0	0	8	8	2	6	3
---	---	---	---	---	---	---

Tabla 2: Representación de la forma en que se almacenan los dígitos

Si bien la suma no se ve afectada por el orden, al desplegar los resultado hace falta aplicar las siguientes líneas de código:

```

1  cout<<"El factorial de "<<num<<" es: ";
2
3  reverse(auxiliar.begin(),auxiliar.end());
4  vector<int>::iterator it=auxiliar.begin();
5
6  for(it;it!=auxiliar.end();it++)
7  {
8      cout<<*it;
9  }

```

Una vez hecho esto, se procede a hacer la suma, cuya función es la siguiente:

```

1  int suma = sum();
2  cout<<"\n\nLa suma de sus dÃgitos es: "<<suma<<endl;

1  int sum()
2  {
3      int sum = accumulate(auxiliar.begin(), auxiliar.end(), 0);
4      return sum;
5  }

```

1.2. Resultados



Fig. 2: Resultados obtenidos del cálculo del factorial y de la respectiva suma de sus dígitos.