

## Ejercicio 1, Contador ASCII

**Disponible desde:** viernes, 23 de octubre de 2020, 12:00

**Límite de entrega:** viernes, 23 de octubre de 2020, 14:10

**Ficheros requeridos:** tipoCaracter.cpp ([Descargar](#))

**Tipo de trabajo:** Individual

Desarrolle un algoritmo en C++ que tome como entrada 10 caracteres y los clasifique en: **Dígitos** (valores ASCII del 48 al 57), **Letras** (valores ASCII del 65 al 122) y **Caracteres especiales** (el resto de los valores ASCII, en este grupo se incluyen la ñ, la Ñ y las vocales tildadas); la salida del programa es la cantidad de caracteres de cada tipo.

=====ejemplo 1=====

entrada: 4 \$ ; W n ñ 2 = y /

```
Dígitos 2
Letras 3
Caracteres especiales 5

<< El programa ha finalizado: código de salida: 0 >>
<< Presione enter para cerrar esta ventana >>
```

=====ejemplo 2=====

```
#include<iostream>
using namespace std;
int main(){
    string cadenIngresada;
    int digitos =0, letras =0, especial =0, codigo=0, cant = 0;
    cin>>cadenIngresada;
    cant = cadenIngresada.length();
    for(int i =0; i < cant; i++)
    {
        codigo = cadenIngresada[i];
        if(codigo>47 && codigo<58)
        {
            digitos++;
        }
        else if(codigo>64 && codigo<123)
        {
            letras++;
        }
        else
        {
            especial++;
        }
    }
    cout<<" Dígitos: "<<digitos<<endl;
    cout<<" Letras: "<<letras<<endl;
    cout<<" Caracteres Especial : "<<especial<<endl;
    return 0;
}
```

```

1 #include<iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     string cadenIngresada;
8     int digitos =0, letras =0, especial =0, codigo=0, cant = 0;
9     cin>>cadenIngresada;
10    cant = cadenIngresada.length();
11    for(int i =0; i < cant; i++)
12    {
13
14        codigo = cadenIngresada[i];
15        if(codigo>47 && codigo<58)
16        {
17            digitos++;
18        }
19        else if(codigo>64 && codigo<123)
20        {
21            letras++;
22        }
23        else
24        {
25            especial++;
26        }
27    }
28    cout<<" Digitos: "<<digitos<<endl;
29    cout<<" Letras: "<<letras<<endl;
30    cout<<" Caracteres Especial : "<<especial<<endl;
31    return 0;
32
33
34

```

```

4$;wnn2=y/
Digitos: 2
Letras: 3
Caracteres Especial : 5

```

Process returned 0 (0x0) execution time : 21.997 s  
Press any key to continue.

## Ejercicio 2, Funcion particionada

### ejercicio 2B

**Disponible desde:** viernes, 23 de octubre de 2020, 12:00

**Límite de entrega:** viernes, 23 de octubre de 2020, 14:00

**Ficheros requeridos:** prueba.cpp ([Descargar](#))

**Tipo de trabajo:** Individual

Dada la función seccionada  $f(x)$  definida de la forma siguiente:

$$f(x) = \begin{cases} -3 & x \leq -2 \\ 4x^4 - 2x^2 + 1 & -2 < x < 0 \\ \frac{x+2}{x} & x \geq 0 \end{cases}$$

Hacer un programa que lea  $n$  números y calcule el valor numérico de los  $n$  números

```
Escriba el valor de n
3
escriba un numero real
-4
f(-4)=-3
escriba un numero real
4
f(4)=1.5
escriba un numero real
0
ND
```

```
#include<iostream>
#include <math.h>
using namespace std;
float f(int x);
int main()
{
    int numN=0, numR = 0 ;
    cout<<"Ingrese el valor de n"<<endl;
    cin>>numN;
    for(int i=0; i<numN; i++)
    {
        cout<<"Escriba un numero real"<<endl;
        cin >>numR;
        if(numR == 0)
            cout<<"ND";
        else
            cout<<"f("<<numR<<")="<<f(numR);
    }
    return 0;
}

float f(int x)
{
    float resultado = 0;
    if(x<= -2)
    {
        return -3;
    }
    else if(x>-2 && x<0)
    {
        return 4*pow(x,4) - 2*pow(x,2) +1;
    }
    else
    {
        resultado = (x+2);
    }
}
```

```

    resultado = (x+2)/ x;
    return resultado;
}
}

```

```

1 #include<iostream>
2 #include <math.h>
3 using namespace std;
4 float f(int x);
5 int main()
6 {
7
8     int numN=0, numR = 0 ;
9     cout<<"Ingrese el valor de n"<<endl;
10    cin>>numN;
11
12    for(int i=0; i<numN; i++)
13    {
14        cout<<"Escriba un numero real"<<endl;
15        cin>>numR;
16        if(numR == 0)
17            cout<<"ND"<<endl;
18        else
19            cout<<"f("<<numR<<")="<<f(numR)<<endl;
20    }
21
22    return 0;
23
24 }

```

```

Ingrese el valor de n
3
Escriba un numero real
-4
f(-4)=-3
Escriba un numero real
4
f(4)=1.5
Escriba un numero real
0
ND
Process returned 0 (0x0)   execution time : 25.786 s
Press any key to continue.

```

```

16     if(numR == 0)
17         cout<<"ND"<<endl;
18     else
19         cout<<"f("<<numR<<")="<<f(numR)<<endl;
20 }
21
22 return 0;
23
24 }
25 float f(int x)
26 {
27     float resultado = 0;
28     if(x<= -2)
29     {
30         return -3;
31     }
32     else if(x>-2 && x<0)
33     {
34         return 4*pow(x,4) - 2*pow(x,2) +1;
35     }
36     else
37     {
38         resultado = (x+2);
39
40         return resultado / x;
41     }
42 }
43

```

```

Ingrese el valor de n
3
Escriba un numero real
-4
f(-4)=-3
Escriba un numero real
4
f(4)=1.5
Escriba un numero real
0
ND
Process returned 0 (0x0)   execution time : 25.786 s
Press any key to continue.

```

### Ejercicio 3. Determinar poblacion

Un modelo de población mundial, en miles de millones de personas, está dado por la ecuación

$$Poblacion = 6.0e^{0.02t},$$

donde  $t$  es el tiempo en años ( $t = 0$  representa enero de 2000,  $t = 1$  representa enero de 2001,  $t = 2$  representa enero de 2002, etc). Usando esta formula escriba un programa que despliegue la población anual desde un año A hasta un año B. Verifique que B sea mayor o igual que A y que A sea mayor o igual que el año 2000, si no se cumplen estas condiciones muestre el siguiente mensaje: Datos no validos.

A continuación se muestran ejemplos de las entradas y salidas para el programa:

==== Ejemplo 1 =====

Entrada: 2000

Entrada: 2004

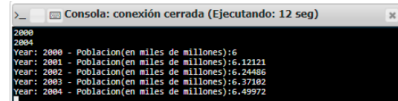
Salida: Year: 2000 - Poblacion(en miles de millones):6

Salida: Year: 2001 - Poblacion(en miles de millones):6.12121

Salida: Year: 2002 - Poblacion(en miles de millones):6.24486

Salida: Year: 2003 - Poblacion(en miles de millones):6.37102

Salida: Year: 2004 - Poblacion(en miles de millones):6.49972



```
>_ Console: conexión cerrada (Ejecutando: 12 seg)
2000
2004
Year: 2000 - Poblacion(en miles de millones):6
Year: 2001 - Poblacion(en miles de millones):6.12121
Year: 2002 - Poblacion(en miles de millones):6.24486
Year: 2003 - Poblacion(en miles de millones):6.37102
Year: 2004 - Poblacion(en miles de millones):6.49972
```

```
#include<iostream>
#include <math.h>
using namespace std;

float poblacion(int t);
int main()
{
    int Tie=0, yearA, yearB;

    cin>> yearA;
    cin >>yearB;
    if(yearA<2000 || yearB<yearA )
    {
        cout<<"Datos invalidos"<<endl;
    }
    else
    {
        do
        {
            cout<<"Year:"<<yearA<<" - Poblacion(en miles de millones): "<<poblacion(Tie)<<endl;
            yearA++;
            Tie++;
        }
        while(yearA<=yearB);
    }

    return 0;
}

float poblacion(int t)
{
    double euler = 2.71828182845904523536;
    return 6*pow(euler,0.02*t );
}
```

```

#include<iostream>
#include<math.h>
using namespace std;
float poblacion(int t);
int main()
{
    int Tie=0, yearA, yearB;
    cin>>yearA;
    cin>>yearB;
    if(yearA<2000 || yearB<yearA )
    {
        cout<<"Datos invalidos"<<endl;
    }
    else
    {
        do
        {
            cout<<"Year:"<<yearA<<" - Poblacion(en miles de millones): "<<poblacion(Tie)<<endl;
            yearA++;
            Tie++;
        }
        while(yearA<=yearB);
    }
    return 0;
}

float poblacion(int t)

```

```

2000
2004
Year:2000 - Poblacion(en miles de millones): 6
Year:2001 - Poblacion(en miles de millones): 6.12121
Year:2002 - Poblacion(en miles de millones): 6.24486
Year:2003 - Poblacion(en miles de millones): 6.37102
Year:2004 - Poblacion(en miles de millones): 6.49972
Process returned 0 (0x0)   execution time : 4.749 s
Press any key to continue.

```

```

int Tie=0, yearA, yearB;
cin>>yearA;
cin>>yearB;
if(yearA<2000 || yearB<yearA )
{
    cout<<"Datos invalidos"<<endl;
}
else
{
    do
    {
        cout<<"Year:"<<yearA<<" - Poblacion(en miles de millones): "<<poblacion(Tie)<<endl;
        yearA++;
        Tie++;
    }
    while(yearA<=yearB);
}
return 0;
}

float poblacion(int t)
{
    double euler = 2.71828182845904523536;
    return 6*pow(euler,0.02*t );
}

```

```

2000
2004
Year:2000 - Poblacion(en miles de millones): 6
Year:2001 - Poblacion(en miles de millones): 6.12121
Year:2002 - Poblacion(en miles de millones): 6.24486
Year:2003 - Poblacion(en miles de millones): 6.37102
Year:2004 - Poblacion(en miles de millones): 6.49972
Process returned 0 (0x0)   execution time : 4.749 s
Press any key to continue.

```