



## SOFTWARE PARA INGENIERÍA (EL227)

### Práctica-02-Solucionario

2017-2 – Sección EL20

#### Pregunta 1 (5 puntos)

Escriba un programa que permita leer una cadena validando que ésta no contenga un espacio en blanco al inicio, en caso eso suceda se deberá pedir nuevamente su ingreso, caso contrario saldrá el mensaje “Cadena correcta”.

Ejemplo:

```
Ingrese una cadena sin espacio en blanco al inicio: programar
Error, ingrese otra vez la cadena

Ingrese una cadena sin espacio en blanco al inicio:  lenguaje C++
Error, ingrese otra vez la cadena

Ingrese una cadena sin espacio en blanco al inicio:Arriba Peru

Cadena correcta
```

Solución:

```
#include <iostream>

using namespace std;

int main()
{
    string cadena;

    //Lectura y validacion de la cadena:
    do
    {
        cout<<"Ingrese una cadena sin espacio en blanco al inicio:";
        getline(cin,cadena);
        if(cadena[0]==' ')
            cout<<"Ingrese, otra vez la cadena\n\n";
    }while(cadena[0]==' '); //si el primer caracter de la cadena es espacio en blanco
                          //volver a leer

    cout<<"\nCadena correcta";

    return 0;
}
```

## Pregunta 2 (5 puntos)

Escriba un programa que permita calcular la suma de los  $n$  elementos de la siguiente serie:

$$\frac{1!}{F(1)^2 + 1} + \frac{2!}{F(2)^2 + 1} + \frac{3!}{F(3)^2 + 1} + \dots + \frac{n!}{F(n)^2 + 1}$$

Donde  $F(x) = (x^2 + 2)/(2x+1)$

El valor de  $n$  será leído por el programa validándose que solamente puede ser un valor positivo no mayor a 15, caso contrario se pedirá nuevamente su ingreso.

El desarrollo de su programa deberá usar **OBLIGATORIAMENTE** las siguientes funciones:

**float** F(**float** x) , **double** factorial(**int** num) , **double** calcularSumaSerie(**int** n)

Ejemplo:

```
Ingrese la cantidad de terminos de la serie: 16
Error, ingrese nuevamente

Ingrese la cantidad de terminos de la serie: 10
La sumatoria de la serie es: 4.03791e+006 ← 168640
```

Solución:

```
#include <iostream>
#include <cmath>

using namespace std;

//funciones:
float f(float x)
{
    return (pow(x,2) + 2)/(2*x+1);
}

double factorial(int num)
{
    double fact=1;
    for(int i=1;i<=num;i++)
    {
        fact = fact*i;
    }
    return fact;
}

double calcularSumaSerie(int n) {
    double acum = 0;

    for(int i=1; i<=n;i++){
        //acumula suma de la serie
        acum = acum + factorial(i)/(pow(f(i),2)+1);
    }

    return acum; //retorna el valor de la suma
}

int main() {
    int n;

    //lectura y validacion de la cantidad de elementos de la serie:
    do{
        cout<<"Ingrese la cantidad de elementos de la serie: ";
        cin>>n;
        if(n<=0 || n>15)
            cout<<"Error, ingrese nuevamente\n\n";
    }while(n<=0 || n>15);

    //se invoca a la funcion que calcula la suma de la serie
    cout<<endl<<"La sumatoria de la serie es: "<<calcularSumaSerie(n);

    return 0;
}
```

### Pregunta 3 (5 puntos)

Escriba un programa que permita lo siguiente:

- Declarar un arreglo unidimensional para almacenar como máximo N valores enteros. N será leído validándose que no sea un valor menor a 10 o mayor a 50, en caso eso suceda, se pedirá nuevamente su ingreso.
- Poblar los 4 primeros valores del arreglo con zeros (0).
- Poblar el resto de valores del arreglo con números aleatorios de 3 cifras a excepción del número 666.
- Mostrar en pantalla todos los elementos del arreglo.

Ejemplo:

```
Ingrese la cantidad de elementos del arreglo: 60
Error, vuelva a ingresar la cantidad

Ingrese la cantidad de elementos del arreglo: 15

Arreglo formado:
0
0
0
0
335
472
716
513
930
876
555
730
272
179
887
```

Solución:

```
#include <iostream>
#include <ctime>
#include <cstdlib>

using namespace std;

int main() {
    srand(time(NULL));

    int N;

    //Lectura y validacion del tamaño del arreglo:
    do
    {
        cout<<"Ingrese la cantidad de elementos del arreglo: ";
        cin>>N;
        if(N<10 || N>50)
            cout<<"Error, vuelva a ingresar la cantidad\n\n";
    }while(N<10 || N>50);

    //declarando al arreglo unidimensional:
    int arreglo[N];

    //poblando los primeros 4 elementos del arreglo con 0: (puede usar for)
    arreglo[0] = 0; arreglo[1] = 0; arreglo[2] = 0; arreglo[3] = 0;

    //poblando los demas valores del arreglo:
    for(int i=4;i<N;i++)
    {
        //volver a asignar valor al arreglo si es que este sale 666
        do{
            arreglo[i] = rand()%(999-100+1)+100;
        }while(arreglo[i]==666);
    }

    //mostrando todos los datos del arreglo:
    cout<<endl<<"Arreglo formado: "<<endl;
    for(int i=0;i<N;i++)
    {
        cout<<arreglo[i]<<endl;
    }

    return 0;
}
```

#### Pregunta 4 (5 puntos)

Escriba un programa que permita leer una cadena formada por nombres de personas separados por un guion. Luego de ello, el programa deberá detectar de la cadena leída los nombres que contengan más de 3 caracteres que no sean vocales, **almacenando en un arreglo unidimensional los nombres detectados** y mostrándolos en pantalla.

**Sugerencia:** El tamaño del arreglo cuando sea declarado puede ser igual a la cantidad total de nombres de la cadena leída.

Ejemplo:

```
Ingrese nombres separados por -: juan-carmen-richard-ana-emilio-judith-marcelo

Nombres detectados con mas de 3 caracteres (no vocales):

carmen
richard
judith
marcelo
```

Solución:

```
#include <iostream>

using namespace std;

//funcion que cuenta la cantidad de caracteres (no vocales) de una cadena
int cuentaCaracteres(string str) {
    int cont=0;

    for(int j=0; j<str.length();j++) {
        //si se detecta en la cadena un caracter (no vocal), incrementa contador
        if(str[j]!='a' && str[j]!='e' && str[j]!='i' && str[j]!='o' && str[j]!='u')
            cont++;
    }

    return cont; //se retorna el contador
}

int main() {

    string cadena, aux;
    int cont=0, p=0, x, k=0;

    //lectura de la cadena:
    cout<<"Ingrese nombres separados por -: ";
    getline(cin,cadena);

    for(int i=0;i<cadena.length();i++){
        if(cadena[i]=='-') //se cuentan los guiones de la cadena
            cont++;
    }

    //se declara el arreglo que contendrá los nombres detectados:
    string nombres[cont+1]; //si hay cont guiones, entonces hay cont+1 nombres

    //proceso de detección de nombres:
    cadena = cadena + "-"; //se agrega un - al final de la cadena

    cout<<endl<<"Nombres detectados con mas de 3 caracteres (no vocales): \n";
    for(int i=1;i<=cont+1;i++) {

        x = cadena.find("-",p); //se busca la posición de un guion
                                //desde la posición p

        //se obtiene el nombre antes del guion
        aux = cadena.substr(p,x-p);

        if(cuentaCaracteres(aux)>3) { //si el nombre tiene mas de 3 caracteres (no vocal)

            cout<<aux<<endl; //se muestra el nombre
            nombres[k] = aux; //se asigna el nombre al arreglo
            k++; //aumenta índice para el siguiente nombre
        }

        p= x+1; //se actualiza posición de búsqueda del guion
    }

    return 0;
}
```