UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE CIENCIAS
Escuela Profesional de Ciencia de la Computación

2

Fundamentos del Programación

2° Practica Calificada - Respuestas

Fecha: 07.10.2025
Tiempo: 2h 30'

Código 20250541C **Apellidos y Nombres :Taipe Ventura Dennis Yudmer**

Procedimiento de Entrega de la Prueba:

Presentar el presente formato consignando su Código, apellidos y nombres, capturas de pantalla que demuestre la resolución de cada uno de los enunciados (código y ejecucion). Publíquelo en su cuenta Github (modo privado) brindando acceso como colaborador a la cuenta **zcodlab**
Nombre del Proyecto: **252CC112C<20250541C>PC2**

De tener inconvenientes con el uso de Github, comprima el código fuente, el presente formato y envíelo al correo: zmamanir@uni.edu.pe

Consignar el Link(Enlace) de Github como respuesta a la Tarea aperturada en el Aula Virtual. Omitir este paso, si lo envió por correo.

**Si se detecta 2 o más trabajos iguales o uso de alguna IA se anula a todos los implicados*

Ejercicio1:

```cpp
#include<iostream>
#include<stdlib.h>
#include<ctime>
using namespace std;
int contar_primos(int n, int q);
int main(){
    srand(time(NULL));
    int n,i,j,z=0;
    cout<<"Ingrese la dimencion de la matriz cuadrada :"<<endl;
    cin>>n;
    int matriz[n][n];
    for(i=0; i<n; i++){
        for(j=0 ; j<n; j++){
            matriz[i][j]=rand()%20;
        }
    }
    for(i=0; i<n; i++){
        cout<<"[ ";
        for(j=0 ; j<n; j++){
            cout<<" "<<matriz[i][j]<<" ";
        }
        cout<<" ]"<<endl;
    }
    int a,b;
    for(i=0; i<n; i++){
        for(j=0 ; j<n; j++){
            a=matriz[i][j];
            b=contar_primos(a,a);
            if(a==0){
                z=z+1;
            }
            else{
                if(b==a-2){
                    z=z+1;
                }
                else{
                    z=z+0;
                }
            }
        }
    }
```

```
    int s=0;
    int d=0;
    for(i=0; i<n; i++){
        for(j=0 ; j<n; j++){
            if(i==j){
                s=s+matriz[i][j];
            }
            else{
                s=s;
            }
            if((i+j+2)>n+1){
                d=d+matriz[i][j];
            }
            else{
                d=d;
            }
        }
    }
    //la funcion "contar_primos" solo determina si un numero es primo mediante indicadores,
    // pero no toma en cuenta si el numero se repite en la matriz.
    cout<<"La matriz tiene :"<<z<<" numeros primos :"<<endl;
    cout<<"La suma de los elemntos de la diagonal principal es: "<<s<<endl;
    cout<<"La suma debajo de la diagonal secundaria es: "<<d;
    return 0;
}
int contar_primos(int n, int q){
    if(q==1){
        return 0;
    }
    else{
        if(n%q==0){
            return 0+contar_primos(n,q-1);
        }
        else{
            return 1+contar_primos(n,q-1);
        }
    }
}
```

Ejercicio2:

Ejercicio3:

```cpp
#include<iostream>
#include<cmath>
double desviacion_estandar( int *m , int i, double z);
using namespace std;
int main(){
    int d;
    int lista[]={20,8,13,19,17,16,12,0,19,20};
    d=sizeof(lista)/sizeof(lista[0]);
    int *m= new int[d];
    double lista2[d];
    int i;
    cout<<"[ ";
    for(i=0 ; i<d ; i++){
        m[i]=lista[i];
        cout<<" "<<m[i]<<" ";
    }
    cout<<" ]";
    double s=0;
    double promedio;
    for(i=0 ; i<d ; i++){
        s=m[i]+s;
    }
    promedio=s/d;
    cout<<" La media es :"<<promedio<<endl;
    double t;
    t=(desviacion_estandar( m , d-1 , promedio )/d-1);
    double des;
    des=pow(t,(0.5));
    cout<<" desviacion estandar es :"<<des<<"
    for(i=0; i<d ; i++){
        if((m[i]-promedio)<0){
            lista2[i]=(m[i]-promedio)*(-1);
        }
        else{
            lista2[i]=(m[i]-promedio);
        }
    }
    int q=0;
    double menor=lista2[0];
    for(i=0; i<d ; i++){
        if(lista2[i] <= menor){
    }
    int q=0;
    double menor=lista2[0];
    for(i=0; i<d ; i++){
        if(lista2[i] <= menor){
            menor=lista2[i];
            q=i;
        }
        else{
            menor=menor;
            q=q;
        }
    }
    cout<<"El numero que mas cercano a la media es: "<<m[q]<<endl;
    for(i=0; i<d ; i++){
        if(lista2[i]>(m[q]+(promedio/2)) || lista2[i]<(m[q]-(promedio/2))){
            lista2[i]=m[q];
        }
        else{
            lista2[i]=lista[i];
        }
    }
    for(i=0 ; i<d ; i++){
        cout<<" "<<lista2[i]<<" ";
    }
    return 0;
}
double desviacion_estandar(int *m, int i, double z){
    if(i==0){
        return ((m[i]-z)*(m[i]-z));
    }
    else{
        return  ((m[i]-z)*(m[i]-z)+desviacion_estandar( m , i-1 , z));
    }
}
```

no usar operador new

no corresponde a los contenidos

dp de los punteros

Ejercicio4: