

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

Fundamentos del Programación

2° Practica Calificada - Respuestas

Fecha: 07.10.2025

Tiempo: 2h 30'

Código 20250337G

Apellidos y Nombres Falera Castillo Mariana Odalis

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 ejecución). Publíquelo en su cuenta Github (modo privado) brindando acceso como colaborador a la cuenta **zcodlab**

Nombre del Proyecto: 252CC112C<CODIGOESTUDIANTE>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:

```
1 #include <iostream>
2 using namespace std;
3
4 bool esPrimo(int num){
5     if(num < 2){
6         return false;
7     }
8     for(int i = 2; i * i <= num; i++){
9         if(num % i == 0){
10            return false;
11        }
12    }
13    return true;
14 }
15
16 int contarPrimos(int *p, int n){
17     int cont = 0;
18     for(int i = 0; i < n; i++){
19         for(int j = 0; j < n; j++){
20             if(esPrimo(*(p + i*n + j))){
21                 cont++;
22             }
23         }
24     }
25     return cont;
26 }
```

```
28 int contarPares(int *p, int n){
29     int cont = 0;
30     for(int i = 0; i < n; i++){
31         if(*(p + i*n + i) % 2 == 0){
32             cont++;
33         }
34     }
35     return cont;
36 }
37
38 int sumarDiagSec(int *p, int n){
39     int suma = 0;
40     for(int i = 0; i < n; i++){
41         for(int j = 0; j < n; j++){
42             if(i + j > n - 1){
43                 suma += *(p + i*n + j);
44             }
45         }
46     }
47     return suma;
48 }
49 }
```

```

50 int main() {
51     int n;
52     cout << "Ingrese el tamaño de la matriz (n<=5): ";
53     cin >> n;
54
55     int M[n][n];
56     cout << "Ingrese los elementos de la matriz: " << endl;
57     for(int i = 0; i < n; i++)
58         for(int j = 0; j < n; j++){
59             cin >> *(M + i*n + j);
60
61             *p = &M[0][0];
62
63     cout << "Matriz ingresada y direcciones" << endl;
64     for(int i = 0; i < n; i++){
65         for(int j = 0; j < n; j++){
66             cout << *(p + i*n + j) << " (" << (p + i*n + j) << ") ";
67         }
68         cout << endl;
69     }
70
71     cout << "Seleccione operacion:" << endl;
72     cout << "1. Contar primos" << endl;
73     cout << "2. Contar pares en la diagonal principal" << endl;
74     cout << "3. Sumar debajo de la diagonal secundaria" << endl;
75     cout << "Opcion: " << endl;
76
77     int opcion;
78     cin >> opcion;

```

```

79
80     int resultado = 0;
81
82     int (*funcion)(int*, int) = 0;
83
84     if(opcion == 1) {
85         funcion = contarPrimos;
86     }else if(opcion == 2){
87         funcion = contarPares;
88     }else if(opcion == 3){
89         funcion = sumarDiagSec;
90     }else{
91         cout << "Opción inválida";
92         return 0;
93     }
94
95     resultado = funcion(p, n);
96
97     cout << "Resultado: " << resultado << endl;
98     return 0;
99 }
100

```

```

Ingrese el tamaño de la matriz (n<=5): 2
Ingrese los elementos de la matriz:
32
23
3
65
Matriz ingresada y direcciones
32 (0x6ffd40) 23 (0x6ffd44)
3 (0x6ffd48) 65 (0x6ffd4c)
Seleccione operacion:
1. Contar primos
2. Contar pares en la diagonal principal
3. Sumar debajo de la diagonal secundaria
Opcion:
2
Resultado: 1

```

Ejercicio2:

Ejercicio3:

```

1  #include <iostream>
2  using namespace std;
3
4  bool simbolosPermitidos(char c) {
5      char simbolos[] = "!@#$%^&*()_-=.,\";
6      for(int i = 0; simbolos[i] != '\0'; i++){ //\0
7          if(c == simbolos[i]){
8              return true;
9          }
10     }
11     return false;
12 }
13
14 int longitud(char *p){
15     int cont = 0;
16     while(*(p + cont) != '\0'){
17         cont++;
18     }
19     return cont;
20 }
21
22 int EsContraseñaSegura(char *p){
23     int len = longitud(p);
24     if(len < 12){
25         cout << "Posicion: 1 Muy corta (menos de 12 caracteres) ";
26         return 1;
27     }
28
29     bool mayus = false, minus = false, num = false, simb = false;
30

```

use pointers a char

de fin a constante

```

31 for(int i = 0; *(p + i) != '\0'; i++){
32     char c = *(p + i);
33     if(c == '\0'){
34         cout << "Posicion: " << i + 1 << " Tiene espacio en blanco";
35         return i + 1;
36     }
37
38     if(!((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z') || (c >= '0' && c <= '9') || simbolosPermitidos(c))){
39         cout << "Posicion: " << i + 1 << " Caracter no permitido: " << c << " ";
40         return i + 1;
41     }
42
43     if(i > 0 && c == *(p + i - 1)){
44         cout << "Posicion: " << i + 1 << " Dos simbolos " << c << c << " consecutivos";
45         return i + 1;
46     }
47
48     if(c >= 'A' && c <= 'Z'){
49         mayus = true;
50     }else if(c >= 'a' && c <= 'z'){
51         minus = true;
52     }else if(c >= '0' && c <= '9'){
53         num = true;
54     }else if(simbolosPermitidos(c)){
55         simb = true;
56     }
57 }
58

```

Activar Wind

```

59 if(!(mayus && minus && num && simb)){
60     cout << "Posicion: 1 No cumple todas las reglas";
61     return 1;
62 }
63
64 cout << "Posicion: 0 Cumple todas las reglas";
65 return 0;
66 }
67
68 int main() {
69     char contrasena[100];
70
71     cout << "Ingrese contraseña: ";
72     cin >> contrasena;
73
74     EsContrasenaSegura(contrasena);
75
76     return 0;
77 }

```

```

Ingrese contraseña: Dhnso#$98403%!KFN
Posicion: 0 Cumple todas las reglas

```

Ejercicio 3

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main(){
6      int NOTAS[] = {20, 8, 13, 19, 17, 16, 12, 0, 19};
7      const int n = 10;
8
9      double promedio;
10
11      int suma1 = 0;
12      for(int i = 0; i < n; i++){
13          suma1 = suma1 + NOTAS[i];
14      }
15      promedio = suma1/n;
16
17      double x[n];
18      for(int i = 0; i < n; i++){
19          x[i] = NOTAS[i] - promedio;
20      }
21      int desvEst = sqrt(((x[n])*(x[n]))/n);
22
23      cout << "Desviacion Estandar: " << desvEst << endl;
24
25      int valorCentral[n];
26      for(int i = 0; i < n; i++){
27          valorCentral = abs(NOTAS[i] - promedio);
28          if(valorCentral[i] < valorCentral[i + 1]){
29              valorCentral = valorCentral[i];
30          }
31      }
```

```
31
32      int NOTASM[10];
33      for(int i = 0; i < n; i++){
34          if(NOTAS[i] - valorCentral > desvEst/2){
35              swap(NOTAS[i], valorCentral);
36          }
37      }
38
39      cout << "Valor Central: " << valorCentral << endl;
40
41      cout << "Notas originales: " << endl;
42      cout << NOTAS;
43
44      cout << "Notas modificadas: " << endl;
45      cout << NOTASM;
46
47      return 0;
48 }
```