



UNIVERSIDAD NACIONAL DE INGENIERÍA

Facultad de Ciencias

Escuela Profesional de Ciencia de la Computación

Primera Práctica Calificada

CC112

15/04/2025 Tiempo: 2:30 horas

Ciclo: 2025-I

Normas:

1. No compartir respuestas/consultas con sus compañeros a través de chats, redes sociales u otros medios digitales.
2. No se permiten apuntes de clase.
3. Las soluciones serán enviadas a la plataforma y/o a la cuenta de correo del profesor.
4. Todo acto anti-ético será amonestado y registrado en el historial del estudiante.

Apellidos: _____ Nombres: _____

Código: _____ Sección: _____

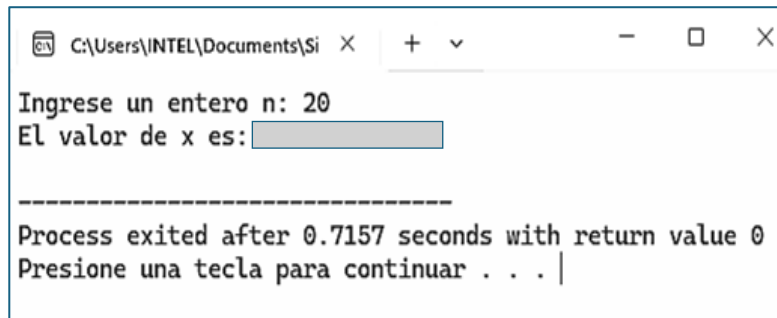
1. [5 puntos] Escriba un programa que solicite un número natural n y calcule el valor de x . Utilice para el cálculo de x una función recursiva

$$\begin{aligned} x &= 1 + \frac{1}{1 - \frac{1}{2 + \frac{1}{3 - \frac{2}{4 + \frac{2}{5 - \frac{3}{6 + \frac{3}{7 - \frac{4}{8 + \frac{4}{9 - \frac{5}{10 + \frac{5}{11 - \frac{6}{12 + \frac{6}{13 - \frac{7}{14 + \frac{7}{\ddots \frac{1}{n}}}}}}}}}}}}}}}}}} \end{aligned}$$

Ejemplo: para $n = 7$

$$\begin{aligned} x &= 1 + \frac{1}{1 - \frac{1}{2 + \frac{1}{3 - \frac{2}{4 + \frac{2}{5 - \frac{3}{6 + \frac{3}{7}}}}} \end{aligned}$$

Ejemplo de ejecución:



```
C:\Users\INTEL\Documents\Si X + - □ X
Ingrese un entero n: 20
El valor de x es: 
-----
Process exited after 0.7157 seconds with return value 0
Presione una tecla para continuar . . . |
```

2. [5 puntos] Dado un arreglo de n números enteros, encuentra el **primer número (de izquierda a derecha) que aparece exactamente una vez**.

Ejemplo1:

$$n = 7$$

$$A = \{4, 5, 1, 2, 1, 4, 2\}$$

Ejemplo 2:

$$n = 8$$

$$A = \{4, 5, 1, 2, 1, 4, 2, 5\}$$

Explicación:

- 4 → aparece 2 veces
- 5 → aparece 1 vez
- 1 → aparece 2 veces
- 2 → aparece 2 veces

Respuesta: -1

Respuesta: 5

3. Se tienen dos vectores cualesquiera de tipo int que inicialmente están desordenados. Usted debe crear las funciones faltantes para que los ordene. La función para ordenar un vector debe tener el siguiente prototipo:
void QuickSort(int *pArray, int n);
El método de ordenamiento debe ser un QuickSort recursivo.

Luego debe hacer otra función Merge que reciba ambos vectores y me devuelva un tercer vector que contenga los elementos de ambos vectores de entrada ordenados y sin repetidos.

La función Merge debe tener el siguiente prototipo:

```
void Merge(int *pVect1, int tam1,      // vector de entrada #1, ordenado de forma ascendente
           int pVect2, int tam2,      // vector de entrada #2, ordenado de forma ascendente
           int *&rOutput, int &nElem // El vector de salida, ordenado de forma ascendente
);
```

El archivo principal (main.cpp) es el siguiente:

```
// main.cpp
int main() {
    int vec1[] = {7, 2, 5, 10};
    int vec2[] = {5, 2, 18, 1, 0, 14, 7, 3, 6};

    QuickSort(vec1, 4);
    QuickSort(vec2, 9);

    int *resultado = nullptr, tam;
    Merge(vec1, 4, vec2, 9, resultado, tam);
    PrintArray(vec1, 4, cout);
    PrintArray(vec2, 9, cout);

    cout << "Resultado: ";
    PrintArray(resultado, tam, cout);
    DestroyArray(resultado);
    return 0;
}
```

- El trabajo es:
Implementar las funciones en archivos separados:
Sorting.cpp
 QuickSort, Merge,
Array.cpp
 PrintArray, imprime de forma flexible en algún stream de tipo
 output.
 DestroyArray: debe borrar el vector y dejarlo en un estado seguro.
- Los prototipos (en archivos .h) deben considerar control para compilacion duplicada.

4. [5 puntos] Una empresa de delivery recibe k pedidos y registra en un arreglo A la distancia en kilómetros a la que debe entregarse cada pedido. La empresa puede contratar como máximo n repartidores que deben llevar grupos de pedidos a repartir.

La empresa desea emplear la menor cantidad de repartidores y que vuelvan los más rápido posible para llevar nuevos pedidos, es decir cada grupo de pedidos debe representar viajes lo más cortos posibles.

Resuelva el problema con el empleo de la búsqueda binaria, mostrando los grupos que hay que formar. Los repartidores solo pueden llevar grupos de pedidos que este registrados en forma consecutiva.

Ejemplo:

$$n = 2$$

$$A = [7, 2, 5, 10, 8]$$

Entonces podríamos formar grupos

- $[7, 2], [5, 10, 8] \rightarrow$ hay un repartidor que viaja 23km
- $[7, 2, 5], [10, 8] \rightarrow$ hay un repartidor que viaja 18km

Por lo tanto, de todos los grupos posibles elegimos la subdivisión $[7, 2, 5], [10, 8]$ pues la distancia máxima es la menor posible.