

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
INE018 MATEMÁTICA COMPUTACIONAL

Examen Parcial
2024-1

Indicaciones generales:

- Duración: 120 minutos.
- No está permitido el uso de ningún material o equipo electrónico adicional al indicado (no celulares, no tablets, no libros).
- **La presentación, la ortografía y la gramática de los trabajos influirán en la calificación.**

Puntaje total: 20 puntos.

Pregunta 1. (2 puntos)

¿Qué caracteres son utilizados para marcar comentarios en un programa en C++?

Pregunta 2. (2 puntos)

Indique cuáles de los siguientes son nombres válidos de variables en C++:

- a. x
- b. formula1
- c. nota_promedio
- d. %correcto
- e. int
- f. entero
- g. hola mundo
- h. unaVariableMuyLarga
- i. 20PuntosEnTotal
- j. nombre-legal
- k. a7x
- l. _nombre_ilegal

Pregunta 3. (2 puntos)

Escriba la forma sintáctica general para cada una de las siguientes estructuras de control: `if`, `while`, `for`. También describa el rol de la sentencia `break`.

Pregunta 4. (2 puntos)

Defina los siguientes términos en relación a las funciones: *llamada*, *argumento*, *retorno*.

Pregunta 5. (2 puntos)

¿Cuál es la diferencia entre un *character* y una *cadena*?

Pregunta 6. (2 puntos)

Dado el siguiente programa:

```
void Imprimir(string y, string z, string x) {
    cout << z << " y " << x << " comen " << y << endl;
}

void Misterio(void) {
    string x = "cpp";
    string y = "Manuel";
    string z = "laptop";
    string futbol = "carapulcra";
    string cpp = "Alejandro";
    Imprimir(x, y, z);
    Imprimir(z, x, y);
    Imprimir("futbol", z, cpp);
    Imprimir(y, futbol, "x");
    Imprimir(y, y, "cpp");
}
```

Escriba la salida de cada una de las llamadas tal como aparecerían en la consola.

Pregunta 7. (2 puntos)

Para cada llamada a la función

```
int Misterio(int a, int b) {
    int c;
    if (a > b) {
        c = a;
    } else if (b % a == 0) {
        c = b;
    } else {
        c = b + (a - (b % a));
    }
    return c;
}
```

escriba el valor retornado:

```
Misterio(4, 2);
Misterio(5, 4);
Misterio(5, 13);
Misterio(5, 17);
Misterio(4, 8);
```

Pregunta 8. (2 puntos)

Dado el siguiente procedimiento

```
void Misterio(int i, int j) {
    while (i != 0 && j != 0) {
        i = i / j;
        j = (j - 1) / 2;
        cout << i << " " << j << " ";
    }
    cout << i << endl;
}
```

escriba la salida de cada una de las siguientes llamadas:

```
Misterio(5, 0);
Misterio(3, 2);
Misterio(16, 5);
Misterio(80, 9);
Misterio(1600, 40);
```

Pregunta 9. (2 puntos)

Escriba una función llamada `TienePuntoMedio` que acepta tres enteros como parámetros y retorna `true` si uno de los enteros es el punto medio entre los otros dos enteros. Tu función debe retornar `false` si no existe tal relación.

Los enteros pueden ser pasados en cualquier orden. El punto medio puede ser el primer, segundo o tercer elemento.

Llamadas como las siguientes deberían retornar `true`:

```
TienePuntoMedio(4, 6, 8);
TienePuntoMedio(2, 10, 6);
TienePuntoMedio(8, 8, 8);
TienePuntoMedio(25, 10, -5);
```

Llamadas como las siguientes deberían retornar `false`:

```
TienePuntoMedio(3, 1, 3);
TienePuntoMedio(1, 3, 1);
TienePuntoMedio(21, 9, 58);
TienePuntoMedio(2, 8, 16);
```

Pregunta 10. (2 puntos)

Escriba un procedimiento llamado **Armonica** que imprima los primeros términos de la siguiente serie:

$$\sum_{k=1}^{\infty} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots$$

Tu procedimiento debe aceptar un número real como parámetro representando un límite, y debe sumar e imprimir los términos de la sumatoria hasta que la suma de términos coincida o exceda el límite.

Por ejemplo, si a tu procedimiento se le pasa 2.0, imprime los términos hasta que la suma sea mayor o igual a 2. Debes redondear tu respuesta tres dígitos luego del punto decimal.

La salida de llamar **Armonica(2.0)** es la siguiente:

$$1 + 1/2 + 1/3 + 1/4 = 2.083$$

Si a tu procedimiento se le pasa un valor menor a 1, no debes imprimir nada en pantalla. Tu salida debe coincidir con el formato solicitado exactamente; note los espacios y signos más separando términos vecinos.

Llamar **Armonica(0.0)** no debe producir salida alguna.

Llamar **Armonica(1.0)** produce la siguiente salida:

$$1 = 1.000$$

Llamar **Armonica(1.5)** produce la siguiente salida:

$$1 + 1/2 = 1.500$$

Llamar **Armonica(2.7)** produce la siguiente salida:

$$1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8 = 2.718$$

Observación: Considerando las sumas parciales $s_n = \sum_{k=1}^n 1/k$ podemos demostrar que para todo n se cumple $s_{2n} - s_n \geq 1/2$ y, por ende, la serie no es convergente. Este resultado nos permite concluir que s_n aumenta ilimitadamente conforme aumenta su número de términos. Sin embargo, esto ocurre de forma muy lenta. De manera precisa, con un poco de conocimiento de cálculo integral es fácil obtener que

$$\ln(n+1) < 1 + \frac{1}{2} \cdots + \frac{1}{n} < \ln(n) + 1$$

para todo n mayor o igual que 2. Así, por ejemplo, son necesarios más de 10000 términos para que la suma sea mayor o igual que 10, y son necesarios más de 10^{43} términos para que la suma sea mayor o igual que 100.

Profesor del curso: Manuel Loaiza Vasquez.

Lima, 4 de junio de 2024.