



**Tercera Práctica Calificada**  
**Ciclo: 2019-1** **Fecha: 29/04/2019**

**CC112**

**Normas:**

1. El alumno entregará esta hoja de examen debidamente llenada con sus datos.
2. Tiempo de prueba: 1:45
3. No se permite: El uso de celulares, internet, USB, ingresar después de 15 min. de iniciado el examen ni salir antes de la hora de finalización.
4. Todo acto anti-ético será amonestado y registrado en el historial del estudiante.

---

Apellidos: \_\_\_\_\_ Nombres: \_\_\_\_\_  
Sección: \_\_\_\_\_ Grupo: \_\_\_\_\_

1. [5 pts.] Escriba un programa que lea cinco números, entre 10 y 100 inclusive. A medida que lea cada número: si es igual a uno leído anteriormente, no haga nada; en caso contrario:  
Guárdelo en un arreglo  
Muestre el conjunto completo de valores únicos introducidos  
En el programa defina el arreglo más pequeño posible para resolver este problema.

Ejemplo de salida:

Ingresar un nro (10-100): 11

11

Ingresar un nro (10-100): 12

11 12

Ingresar un nro (10-100): 12

Ingresar un nro (10-100): 13

11 12 13

Ingresar un nro (10-100): 13

2. [5 pts.] Escriba un programa para simular el lanzamiento de dos dados. Cada dado muestra un valor entero del 1 al 6, por lo que la suma de los valores variará del 2 al 12, siendo 7 la suma más frecuente, mientras que 2 y 12 serán las sumas menos frecuentes. Su aplicación debe tirar los dados 20000 de veces. Utilice un arreglo unidimensional para registrar el número de veces que aparezca cada una de las posibles sumas.

Muestre los resultados en formato tabular, por ejemplo:

2 salió: 518 veces.

3 salió: 1104 veces.

4 salió: 1547 veces.

5 salió: 2244 veces.

6 salió: 2828 veces.

7 salió: 3365 veces.

8 salió: 2830 veces.

9 salió: 2222 veces.

10 salió: 1685 veces.

11 salió: 1120 veces.

12 salió: 537 veces.

**3. [5 ptos.]** Se tiene un montón (=20) de barras de distintos tamaños que no se deben quebrar ni piezar, porque tienen argollas en las puntas. Se hace un inventario de las barras y su tamaño. Escriba un programa para seleccionar 3 barras que permitan formar el triángulo con la mayor superficie para una antena.

Ejemplo de vista del monitor:

Las 20 barras miden:

4 7 **18 16** 14 16 7 13 10 2 3 8 11 **20** 4 7 1 7 13 17

La superficie máxima es: 136.79 m<sup>2</sup>.

Las barras son barra(2)=18, barra(3)=16, barra(13)=20.

Sugerencia de diseño:

- 1) Generar un arreglo **int arr[20]**, con elementos aleatorios entre 1 y 20. No usar el primer `srand()` porque queremos que el montón sea el mismo para todas las corridas.
- 2) Reportar las medidas de las 20 barras del montón
- 3) Iniciar **max=0**; (la superficie máxima)
- 4) Para seleccionar una triada utilice:

```
for(i=0; i<n-2; i++)           // primera barra
    for(j=... )                 // segunda barra
        for(k=j+1; k<n; k++){   // tercera barra
            // Verifica si arr[i], arr[j] y arr[k] forman un triángulo: si
            // Calcular la superficie con una función:
            float superficie(int l1, int l2, int l3);
            // Verificar la condición de máximo, sí:
            guarda max y los 3 elementos
        }
```

- 5) Imprimir la salida final

**4. [5 ptos.]** La cola de un hospital se ordena de mayor a menor gravedad, ejemplo: 10, 6, 4. Cada paciente nuevo que llega debe ingresar a la **mayor brevedad** posible en su puesto. No vamos a considerar las salidas de cola. Escriba un programa muy veloz que mantenga la cola, de 20 pacientes, ordenada y termine cuando ingrese la gravedad = 0.

Ejemplo de vista del monitor:

10 6 4

Ingrese la gravedad del paciente: 8

10 8 6 4

Ingrese la gravedad del paciente: 1

10 8 6 4 1

Ingrese la gravedad del paciente: 0

Sugerencia de un diseño muy veloz para ingresar un dato:

Se tiene: `int arr[20] = {10 6 4}` e ingresa 8=grave

Mientras `grave > (4=arr[2])` → copiar 1 posición a la derecha `arr = 10 6 4 4`

Mientras `grave > (6=arr[1])` → copiar 1 posición a la derecha `arr = 10 6 6 4`

Al final `grave <= (10=arr[0])` → `arr = 10 8 6 4`