**Programación en C**

**FCEN - 2024**

**Práctica unidad 3**

**Estructuras de selección y control**

# Estructuras condicionales

1. Escriba un código que reciba un número ingresado por el usuario, luego indique si dicho número es positivo, negativo o cero, y si es par o impar.
2. En un país se utilizan diferentes tasas impositivas según el sueldo bruto anual del contribuyente. Los contribuyentes casados suman sus ingresos y pagan impuestos sobre el total. La tabla muestra los cálculos de la tasa de impuestos.

| **Categoria** | **Sueldo bruto anual** | **Tasa impositiva** |
| --- | --- | --- |
| A | Menor o igual a 300000 | 3% |
| B | Entre 300000 y 450000 | 8% |
| C | Entre 450000 y 700000 | 13% |
| D | Entre 700000 y 1200000 | 20% |
| E | Más de 1200000 | 35% |

a) Diseñe un proceso de decisión que calcule el impuesto a pagar teniendo en cuenta las siguientes entradas: soltero/a o casado/a, si es casado/a debe indicar dos sueldos, si es soltero/a uno.

b) El código debe indicar en qué categoría está el contribuyente, cuánto impuesto deberá pagar y cuál es el sueldo mensual.

1. Escriba un programa que convierta grados Celsius a Fahrenheit y Fahrenheit a Celsius. El usuario debe indicar la temperatura y la unidad (C para Celsius y F para Fahrenheit) y el código luego debe realizar la conversión y mostrar el resultado.
2. Escriba un programa que solicite al usuario el ingreso de una letra del abecedario, si esta es una vocal debe devolver Verdadero, en cualquier otro caso Falso.
3. ¿Qué errores de sintaxis tiene el siguiente código?

if x > 25.0

y = x

else

y = z;

1. ¿Qué valor tendrá **consumo** si **velocidad** es igual a 120?

if (velocidad > 80)

consumo = 10.00;

else if (velocidad > 100)

consumo = 12.00;

else if (velocidad > 120)

consumo = 15.00;

1. ¿Puede explicar cómo es el funcionamiento de estos dos ejemplos de código? ¿Cuál es la diferencia principal? ¿Qué valor tendrá x al finalizar las dos versiones del código?

| x = 0;  if (x >= 0)  x++;  else if (x >= 1)  x+= 2; | x = 0;  if (x >= 0)  x++;  if (x >= 1)  x+= 2; |
| --- | --- |
| x = \_\_\_ | x = \_\_\_\_ |

# Estructuras repetitivas

1. Diseñe un algoritmo para determinar el número más grande y más pequeño ingresado por el usuario. Debe incluir una variable de corte o centinela en su código, solo se deben permitir números positivos.
2. Tienes tres dados estándar. Calcule cuántas formas existen para lograr un total de 10 entre los tres dados. Luego calcule la probabilidad de tirar un diez.
3. Analice qué hace el siguiente bucle while, luego reescribalo utilizando for y do-while

num = 10;

while (num <= 100) {

printf("%d \n",num);

num += 10;

}

1. Suponiendo que m = 3 y n = 5, indique cuál es la salida de los siguientes dos códigos:

for (i = 0; i < n; i++){

for (j = 0; j < i; j++){

putchar('\*');

}

putchar('\n');

}

for (i = n; i > 0; i--) {

for (j = m; j > 0; j--){

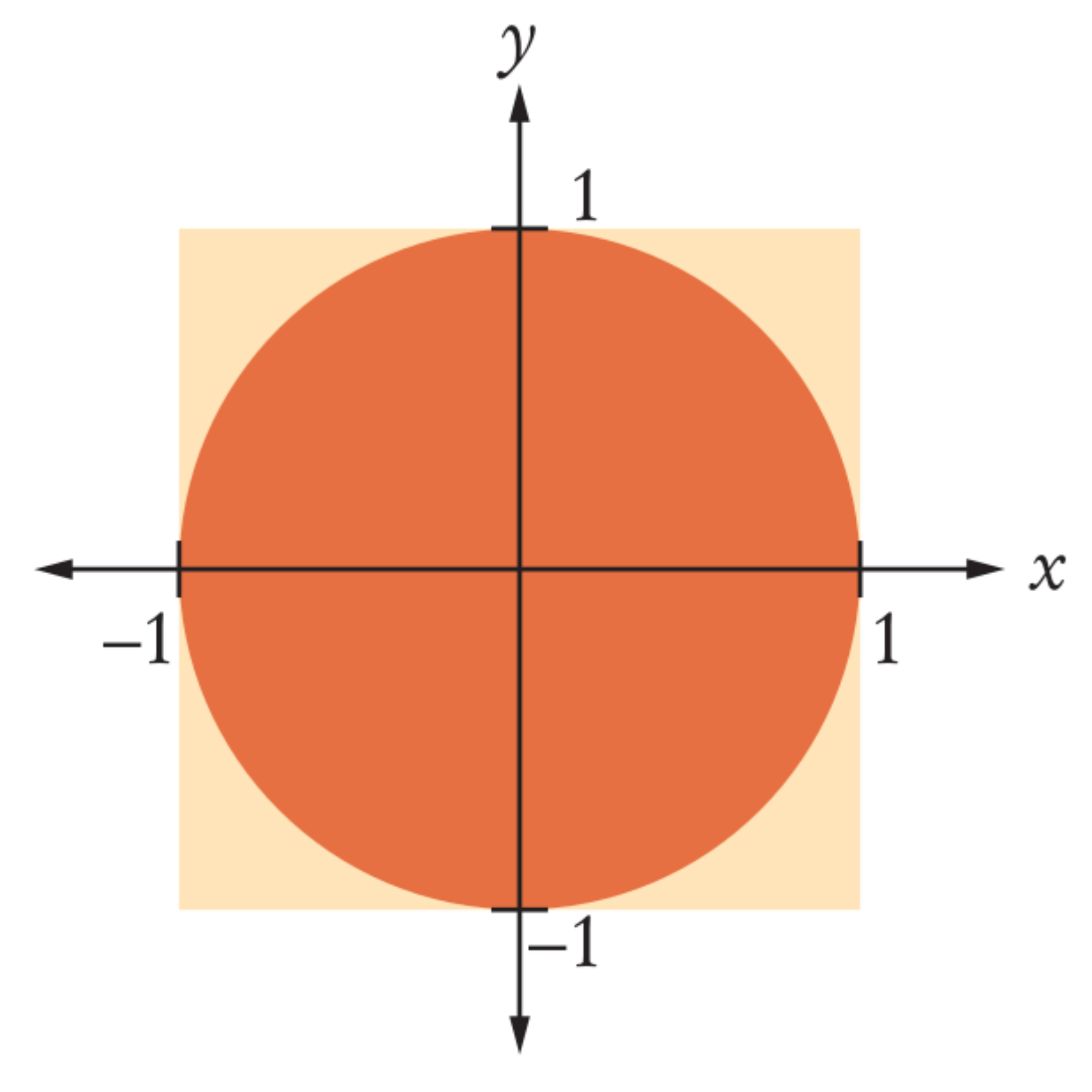
putchar('\*');

}

putchar('\n');

}

1. Escriba un programa que imprima los números divisibles por 7 y 5 que existen entre 1200 y 1450.
2. Un método científico común para estimar valores desconocidos se conoce como el método de Monte Carlo, nombrado por el famoso casino. El concepto básico es usar la aleatoriedad para resolver problemas, en este ejercicio queremos usar el método de Monte Carlo para estimar π.
   1. Usando un programa podemos simular un lanzamiento de dardo en un círculo de radio 1, centrado alrededor (0, 0) como se muestra a continuación.



* 1. Simulamos esto generando una gran cantidad de coordenadas (x, y) contenidas dentro de un cuadrado de longitud de lado = 2.
  2. Luego contamos el número de veces que las coordenadas se encuentran dentro del círculo de radio = 1. Eso es cuando x^2 + y^2 <= 1.
  3. Debido a la aleatoriedad de nuestros lanzamientos, esperamos que la proporción de puntos acertados/intentos sea igual a la proporción de el área del círculo / cuadrado (o, π/4). Por lo tanto, nuestra estimación de π está dada por π = 4.0 \* (aceptado / intentos).
  4. Los números aleatorios entre -1 y 1 se pueden generar de la siguiente manera:

#include <stdlib.h>

float r = -1 + rand() / (float) (RAND\_MAX / 2);

* 1. Diseñar un algoritmo para estimar π a 6 cifras significativas, por 100, 1000 y 10000 intentos, y luego comparar los resultados. Puede probar con valores de intentos mayores, 1 millón, 10 millones, la PC se va a demorar más en realizar el cálculo.
  2. Si realizó el curso de Python, puede hacer una comparación de rendimiento entre esta versión del código Monte Carlo y la versión que desarrolló en Python.

# 

# Extras

1. Escribir un programa que visualice un triángulo isósceles.

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

1. Escriba un programa que genere un triángulo que tenga hasta **n** caracteres de ancho y **(n\*2)-1** caracteres de alto, por ejemplo para **n=5** el codigo deberia mostrar:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*