# Day 10: Miscellaneous (Bitwise, Macros, Preprocessors, etc.)

## 1. Write a program to demonstrate bitwise AND, OR, XOR.

#include <stdio.h>  
int main() {  
 int a = 5, b = 3;  
 printf("AND: %d\n", a & b);  
 printf("OR: %d\n", a | b);  
 printf("XOR: %d\n", a ^ b);  
 return 0;  
}

## 2. Write a program to find whether a number is even using bitwise operator.

#include <stdio.h>  
int main() {  
 int num;  
 printf("Enter a number: ");  
 scanf("%d", &num);  
 if (num & 1)  
 printf("Odd\n");  
 else  
 printf("Even\n");  
 return 0;  
}

## 3. Write a program to toggle a bit of an integer.

#include <stdio.h>  
int main() {  
 int num, pos;  
 printf("Enter number and bit position to toggle: ");  
 scanf("%d %d", &num, &pos);  
 num ^= (1 << pos);  
 printf("Result: %d\n", num);  
 return 0;  
}

## 4. Write a macro to find maximum of two numbers.

#include <stdio.h>  
#define MAX(a, b) ((a) > (b) ? (a) : (b))  
  
int main() {  
 int x = 10, y = 20;  
 printf("Max: %d\n", MAX(x, y));  
 return 0;  
}

## 5. Write a macro to calculate area of a circle.

#include <stdio.h>  
#define PI 3.14159  
#define AREA(r) (PI \* (r) \* (r))  
  
int main() {  
 float radius = 5;  
 printf("Area: %.2f\n", AREA(radius));  
 return 0;  
}

## 6. Write a program to define and use a preprocessor constant.

#include <stdio.h>  
#define LENGTH 10  
  
int main() {  
 printf("Constant LENGTH: %d\n", LENGTH);  
 return 0;  
}

## 7. Write a program to implement a simple calculator using switch.

#include <stdio.h>  
int main() {  
 char op;  
 float a, b;  
 printf("Enter operator (+, -, \*, /): ");  
 scanf(" %c", &op);  
 printf("Enter two operands: ");  
 scanf("%f %f", &a, &b);  
  
 switch(op) {  
 case '+': printf("Result: %.2f\n", a + b); break;  
 case '-': printf("Result: %.2f\n", a - b); break;  
 case '\*': printf("Result: %.2f\n", a \* b); break;  
 case '/': if (b != 0) printf("Result: %.2f\n", a / b);   
 else printf("Division by zero!\n"); break;  
 default: printf("Invalid operator\n");  
 }  
 return 0;  
}

## 8. Write a program to convert binary to decimal.

#include <stdio.h>  
#include <math.h>  
  
int main() {  
 int bin, dec = 0, i = 0;  
 printf("Enter binary number: ");  
 scanf("%d", &bin);  
 while (bin != 0) {  
 dec += (bin % 10) \* pow(2, i);  
 ++i;  
 bin /= 10;  
 }  
 printf("Decimal: %d\n", dec);  
 return 0;  
}

## 9. Write a program to implement a basic stack using array.

#include <stdio.h>  
#define SIZE 5  
int stack[SIZE], top = -1;  
  
void push(int val) {  
 if (top == SIZE - 1)  
 printf("Stack Overflow\n");  
 else  
 stack[++top] = val;  
}  
  
void pop() {  
 if (top == -1)  
 printf("Stack Underflow\n");  
 else  
 printf("Popped: %d\n", stack[top--]);  
}  
  
void display() {  
 if (top == -1)  
 printf("Stack is empty\n");  
 else {  
 for (int i = top; i >= 0; i--)  
 printf("%d ", stack[i]);  
 printf("\n");  
 }  
}  
  
int main() {  
 push(10);  
 push(20);  
 display();  
 pop();  
 display();  
 return 0;  
}

## 10. Write a program to implement a basic queue using array.

#include <stdio.h>  
#define SIZE 5  
int queue[SIZE], front = -1, rear = -1;  
  
void enqueue(int val) {  
 if (rear == SIZE - 1)  
 printf("Queue Full\n");  
 else {  
 if (front == -1) front = 0;  
 queue[++rear] = val;  
 }  
}  
  
void dequeue() {  
 if (front == -1 || front > rear)  
 printf("Queue Empty\n");  
 else  
 printf("Dequeued: %d\n", queue[front++]);  
}  
  
void display() {  
 if (front == -1 || front > rear)  
 printf("Queue Empty\n");  
 else {  
 for (int i = front; i <= rear; i++)  
 printf("%d ", queue[i]);  
 printf("\n");  
 }  
}  
  
int main() {  
 enqueue(10);  
 enqueue(20);  
 display();  
 dequeue();  
 display();  
 return 0;  
}