C Programming Exercises

# Program to swap two values by using call by value and call by address methods

#include <stdio.h>  
  
// Function to swap using call by value  
void swapByValue(int a, int b) {  
 int temp = a;  
 a = b;  
 b = temp;  
}  
  
// Function to swap using call by address (pointers)  
void swapByAddress(int \*a, int \*b) {  
 int temp = \*a;  
 \*a = \*b;  
 \*b = temp;  
}  
  
int main() {  
 int x = 10, y = 20;  
  
 printf("Before swapping (call by value): x = %d, y = %d\n", x, y);  
 swapByValue(x, y);  
 printf("After swapping (call by value): x = %d, y = %d\n", x, y);  
  
 printf("Before swapping (call by address): x = %d, y = %d\n", x, y);  
 swapByAddress(&x, &y);  
 printf("After swapping (call by address): x = %d, y = %d\n", x, y);  
  
 return 0;  
}

# Program to know how the one-dimensional array works using pointers

#include <stdio.h>  
  
int main() {  
 int arr[5] = {10, 20, 30, 40, 50};  
 int \*ptr = arr; // Pointer to the array  
  
 printf("Elements of the array using pointers:\n");  
 for (int i = 0; i < 5; i++) {  
 printf("%d ", \*(ptr + i));  
 }  
  
 return 0;  
}

# Program to read and display the elements of an array in reverse order using pointers

#include <stdio.h>  
  
int main() {  
 int arr[5] = {10, 20, 30, 40, 50};  
 int \*ptr = arr + 4; // Pointer to the last element  
  
 printf("Elements of the array in reverse order using pointers:\n");  
 for (int i = 4; i >= 0; i--) {  
 printf("%d ", \*(ptr - i));  
 }  
  
 return 0;  
}

# Program which demonstrates character pointer

#include <stdio.h>  
  
int main() {  
 char str[] = "Hello, World!";  
 char \*ptr = str; // Pointer to the string  
  
 printf("String using character pointer: %s\n", ptr);  
  
 return 0;  
}

# Program to know how to call a function

#include <stdio.h>  
  
// Function declaration  
void greet();  
  
int main() {  
 greet(); // Function call  
 return 0;  
}  
  
// Function definition  
void greet() {  
 printf("Hello, welcome to function call in C!\n");  
}

# Program to make function declarations before the main and write the definitions outside

#include <stdio.h>  
  
// Function declarations  
void add();  
void subtract();  
  
int main() {  
 add();  
 subtract();  
 return 0;  
}  
  
// Function definitions outside main  
void add() {  
 int a = 5, b = 3;  
 printf("Addition: %d\n", a + b);  
}  
  
void subtract() {  
 int a = 5, b = 3;  
 printf("Subtraction: %d\n", a - b);  
}

# Program to find addition, subtraction, multiplication, and division of two floats using functions

#include <stdio.h>  
  
// Function declarations  
float add(float a, float b);  
float subtract(float a, float b);  
float multiply(float a, float b);  
float divide(float a, float b);  
  
int main() {  
 float x = 12.5, y = 4.5;  
 printf("Addition: %.2f\n", add(x, y));  
 printf("Subtraction: %.2f\n", subtract(x, y));  
 printf("Multiplication: %.2f\n", multiply(x, y));  
 printf("Division: %.2f\n", divide(x, y));  
  
 return 0;  
}  
  
// Function definitions  
float add(float a, float b) {  
 return a + b;  
}  
  
float subtract(float a, float b) {  
 return a - b;  
}  
  
float multiply(float a, float b) {  
 return a \* b;  
}  
  
float divide(float a, float b) {  
 return a / b;  
}

# Program to know about different categories of functions

// In C, there are four categories of functions:  
// 1. Function with arguments and with return value  
// 2. Function with arguments and without return value  
// 3. Function without arguments and with return value  
// 4. Function without arguments and without return value  
  
#include <stdio.h>  
  
// Function with arguments and with return value  
int add(int a, int b) {  
 return a + b;  
}  
  
// Function with arguments and without return value  
void displaySum(int a, int b) {  
 printf("Sum: %d\n", a + b);  
}  
  
// Function without arguments and with return value  
int getNumber() {  
 return 42;  
}  
  
// Function without arguments and without return value  
void greet() {  
 printf("Hello, C Programmer!\n");  
}  
  
int main() {  
 // Category 1  
 int sum = add(3, 4);  
 printf("Sum from add function: %d\n", sum);  
  
 // Category 2  
 displaySum(5, 6);  
  
 // Category 3  
 int number = getNumber();  
 printf("Number from getNumber function: %d\n", number);  
  
 // Category 4  
 greet();  
  
 return 0;  
}

# C program to understand function with arguments and with return value

#include <stdio.h>  
  
// Function with arguments and return value  
int multiply(int a, int b) {  
 return a \* b;  
}  
  
int main() {  
 int result = multiply(5, 3);  
 printf("Multiplication result: %d\n", result);  
 return 0;  
}

# C program to understand function without arguments and without return value

#include <stdio.h>  
  
// Function without arguments and without return value  
void printMessage() {  
 printf("This is a function without arguments and without return value.\n");  
}  
  
int main() {  
 printMessage(); // Function call  
 return 0;  
}