**Assignment 1 Solutions - C Programming**

Question 1: Miles to Kilometers Converter

#include <stdio.h>

// Macro for the conversion factor as requested

#define KM\_PER\_MILE 1.609

int main() {

// Using float for variables since the numbers can have decimals

float miles, kilometers;

// Ask the user for input

printf("Enter the distance in miles: ");

scanf("%f", &miles);

// The actual conversion

kilometers = miles \* KM\_PER\_MILE;

// Print the result

printf("%.2f miles is equal to %.2f kilometers.\n", miles, kilometers);

return 0;

}

**Sample Output:**

**Enter the distance in miles: 10**

**10.00 miles is equal to 16.09 kilometers.**

Question 2: Count Positive, Negative, and Zero Numbers

#include <stdio.h>

int main() {

int total\_count, i, current\_number;

int positive\_count = 0;

int negative\_count = 0;

int zero\_count = 0;

// Find out how many numbers the user wants to check

printf("How many numbers will you enter? ");

scanf("%d", &total\_count);

printf("Please enter %d numbers:\n", total\_count);

// Loop to read each number

for (i = 0; i < total\_count; i++) {

scanf("%d", &current\_number);

// Check the number and update the correct counter

if (current\_number > 0) {

positive\_count++;

} else if (current\_number < 0) {

negative\_count++;

} else {

zero\_count++;

}

}

// Display the final counts

printf("Positive numbers: %d\n", positive\_count);

printf("Negative numbers: %d\n", negative\_count);

printf("Zeros: %d\n", zero\_count);

return 0;

}

**Sample Output:**

**How many numbers will you enter? 5**

**Please enter 5 numbers:**

**10**

**-5**

**0**

**20**

**-2**

**Positive numbers: 2**

**Negative numbers: 2**

**Zeros: 1**

Question 3: Tax Calculation

#include <stdio.h>

int main() {

double salary, tax\_due;

// Get the salary from the user

printf("Enter the salary amount: $");

scanf("%lf", &salary);

// Check all the salary ranges from the table

if (salary < 0.0 || salary > 150000.00) {

tax\_due = -1.0; // Outside the table range

} else if (salary <= 14999.99) {

tax\_due = 0.00 + (salary \* 0.15);

} else if (salary <= 29999.99) {

tax\_due = 2250.00 + (salary - 15000.00) \* 0.18;

} else if (salary <= 49999.99) {

tax\_due = 5400.00 + (salary - 30000.00) \* 0.22;

} else if (salary <= 79999.99) {

tax\_due = 11000.00 + (salary - 50000.00) \* 0.27;

} else { // This covers 80,000.00 to 150,000.00

tax\_due = 21600.00 + (salary - 80000.00) \* 0.33;

}

// Print the final calculated tax

if (tax\_due == -1.0) {

printf("Salary is outside the valid range. Result: -1.0\n");

} else {

printf("The tax due is: $%.2f\n", tax\_due);

}

return 0;

}

**Sample Output:**

**Enter the salary amount: $45000**

**The tax due is: $8700.00**

Question 4: Area Calculator (Menu Driven)

#include <stdio.h>

// PI for circle calculation

#define PI 3.14159

// Function declarations

void calculateSquareArea();

void calculateRectangleArea();

void calculateCircleArea();

int main() {

int choice;

do {

// Print the menu

printf("\n--- Geometry Calculator Menu ---\n");

printf("1. Calculate Area of a Square\n");

printf("2. Calculate Area of a Rectangle\n");

printf("3. Calculate Area of a Circle\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

// Use a switch to call the right function

switch (choice) {

case 1:

calculateSquareArea();

break;

case 2:

calculateRectangleArea();

break;

case 3:

calculateCircleArea();

break;

case 4:

printf("Exiting program. Goodbye!\n");

break;

default:

printf("Invalid choice! Please enter a number between 1 and 4.\n");

}

} while (choice != 4); // Keep looping until the user chooses to exit

return 0;

}

void calculateSquareArea() {

float side, area;

printf("Enter the side of the square: ");

scanf("%f", &side);

area = side \* side;

printf("Area of the square is: %.2f\n", area);

}

void calculateRectangleArea() {

float length, width, area;

printf("Enter the length of the rectangle: ");

scanf("%f", &length);

printf("Enter the width of the rectangle: ");

scanf("%f", &width);

area = length \* width;

printf("Area of the rectangle is: %.2f\n", area);

}

void calculateCircleArea() {

float radius, area;

printf("Enter the radius of the circle: ");

scanf("%f", &radius);

area = PI \* radius \* radius;

printf("Area of the circle is: %.2f\n", area);

}

**Sample Output:**

**--- Geometry Calculator Menu ---**

**1. Calculate Area of a Square**

**2. Calculate Area of a Rectangle**

**3. Calculate Area of a Circle**

**4. Exit**

**Enter your choice: 2**

**Enter the length of the rectangle: 10**

**Enter the width of the rectangle: 5**

**Area of the rectangle is: 50.00**

**--- Geometry Calculator Menu ---**

**1. Calculate Area of a Square**

**2. Calculate Area of a Rectangle**

**3. Calculate Area of a Circle**

**4. Exit**

**Enter your choice: 4**

**Exiting program. Goodbye!**

Question 5: Fibonacci Series

#include <stdio.h>

int main() {

int n, i;

// t1 and t2 are the first two terms

int t1 = 0, t2 = 1;

// next\_term holds the sum of the previous two

int next\_term = t1 + t2;

printf("Enter the number of Fibonacci elements to display: ");

scanf("%d", &n);

// Print the first two terms

printf("Fibonacci Series: %d, %d", t1, t2);

// Loop to generate and print the rest of the terms

// We start from i = 3 because the first two are already printed

for (i = 3; i <= n; ++i) {

printf(", %d", next\_term);

t1 = t2;

t2 = next\_term;

next\_term = t1 + t2;

}

printf("\n");

return 0;

}

**Sample Output:**

**Enter the number of Fibonacci elements to display: 8**

**Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13**

Question 6: Print Multiplication Tables from X to Y

#include <stdio.h>

int main() {

int table\_x, table\_y, i, j;

printf("Enter the starting table number (X): ");

scanf("%d", &table\_x);

printf("Enter the ending table number (Y): ");

scanf("%d", &table\_y);

// Outer loop for each table from X to Y

for (i = table\_x; i <= table\_y; i++) {

printf("\n--- Multiplication Table for %d ---\n", i);

// Inner loop for multipliers 1 to 10

for (j = 1; j <= 10; j++) {

printf("%d x %d = %d\n", i, j, i \* j);

}

}

return 0;

}

**Sample Output:**

**Enter the starting table number (X): 3**

**Enter the ending table number (Y): 4**

**--- Multiplication Table for 3 ---**

**3 x 1 = 3**

**3 x 2 = 6**

**3 x 3 = 9**

**3 x 4 = 12**

**3 x 5 = 15**

**3 x 6 = 18**

**3 x 7 = 21**

**3 x 8 = 24**

**3 x 9 = 27**

**3 x 10 = 30**

**--- Multiplication Table for 4 ---**

**4 x 1 = 4**

**4 x 2 = 8**

**4 x 3 = 12**

**4 x 4 = 16**

**4 x 5 = 20**

**4 x 6 = 24**

**4 x 7 = 28**

**4 x 8 = 32**

**4 x 9 = 36**

**4 x 10 = 40**

Question 7: Iterative Factorial

#include <stdio.h>

int main() {

int n, i;

// Use long long because factorials get big very quickly!

long long factorial = 1;

printf("Enter an integer to find its factorial: ");

scanf("%d", &n);

// Factorial is not defined for negative numbers

if (n < 0) {

printf("Error! Factorial of a negative number doesn't exist.\n");

} else {

// Loop from 1 to n, multiplying each time

for (i = 1; i <= n; ++i) {

factorial = factorial \* i;

}

printf("Factorial of %d = %lld\n", n, factorial);

}

return 0;

}

**Sample Output:**

**Enter an integer to find its factorial: 6**

**Factorial of 6 = 720**

Question 8: Swap Two Numbers Using a Function

#include <stdio.h>

// The function takes pointers so it can modify the original variables

void swap(int \*num1\_ptr, int \*num2\_ptr) {

int temp;

// Store the value of the first number in a temporary variable

temp = \*num1\_ptr;

// Put the second number's value into the first

\*num1\_ptr = \*num2\_ptr;

// Put the stored temp value into the second number

\*num2\_ptr = temp;

}

int main() {

int num1, num2;

printf("Enter first number: ");

scanf("%d", &num1);

printf("Enter second number: ");

scanf("%d", &num2);

printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);

// Pass the memory addresses of the variables to the function

swap(&num1, &num2);

printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);

return 0;

}

**Sample Output:**

**Enter first number: 10**

**Enter second number: 20**

**Before swapping: num1 = 10, num2 = 20**

**After swapping: num1 = 20, num2 = 10**

Question 9: Get an Integer Within a Range

#include <stdio.h>

// This function will keep asking for a number until it's in the valid range

int getNumberInRange(int n\_min, int n\_max) {

int user\_input;

do {

printf("Please enter an integer between %d and %d: ", n\_min, n\_max);

scanf("%d", &user\_input);

if (user\_input < n\_min || user\_input > n\_max) {

printf("Invalid input. Please try again.\n");

}

} while (user\_input < n\_min || user\_input > n\_max); // Loop if number is outside range

return user\_input;

}

int main() {

int min\_val = 10;

int max\_val = 20;

int valid\_number;

printf("We need a number for our program.\n");

// Call the function to get a valid number

valid\_number = getNumberInRange(min\_val, max\_val);

printf("Thank you! You entered the valid number: %d\n", valid\_number);

return 0;

}

**Sample Output:**

**We need a number for our program.**

**Please enter an integer between 10 and 20: 5**

**Invalid input. Please try again.**

**Please enter an integer between 10 and 20: 25**

**Invalid input. Please try again.**

**Please enter an integer between 10 and 20: 15**

**Thank you! You entered the valid number: 15**

Question 10: Nested Loop Pattern

#include <stdio.h>

int main() {

int i, j;

// --- Part 1: Print the top half of the pattern (increasing) ---

for (i = 0; i <= 5; i++) {

// Inner loop prints numbers from 0 up to the current row number 'i'

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

printf("%d", j);

}

printf("\n"); // Move to the next line after each row

}

// --- Part 2: Print the bottom half of the pattern (decreasing) ---

// Outer loop counts down from 4 to 0

for (i = 4; i >= 0; i--) {

// Inner loop prints numbers from 0 up to the current row number 'i'

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

printf("%d", j);

}

printf("\n"); // Move to the next line

}

return 0;

}

**Sample Output:**

**0**

**01**

**012**

**0123**

**01234**

**012345**

**01234**

**0123**

**012**

**01**

**0**