**Assignment No-7**

1. Write a program, which accepts a character from the user and checks if it is an alphabet, digit or punctuation symbol. If it is an alphabet, check if it is uppercase or lowercase and then change the case.

#include <stdio.h> // for input/output functions

#include <ctype.h> // for character checking functions like isalpha, isdigit, etc.

int main() {

char ch;

printf("Enter a character: ");

scanf("%c", &ch); // input character

if (isalpha(ch)) { // check if it's an alphabet (A-Z or a-z)

printf("It is an alphabet.\n");

if (isupper(ch)) { // check if uppercase

printf("It is uppercase. Changing to lowercase: %c\n", tolower(ch));

} else { // otherwise, it's lowercase

printf("It is lowercase. Changing to uppercase: %c\n", toupper(ch));

}

} else if (isdigit(ch)) { // check if it's a digit (0-9)

printf("It is a digit.\n");

} else { // if not alphabet or digit, it's punctuation/symbol

printf("It is a punctuation/symbol.\n");

}

return 0; // end program

}

2. Write a menu driven program to perform the following operations till the user selects Exit. Accept appropriate data for each option. Use standard library functions from math.h i. Sine ii. Cosine iii. log iv. ex v. Square Root vi. Exit

#include <stdio.h> // input/output

#include <math.h> // math functions like sin, cos, log, exp, sqrt

int main() {

int choice;

double value, result;

do {

printf("\n--- Math Operations Menu ---\n");

printf("1. Sine\n2. Cosine\n3. Logarithm (log base e)\n4. e^x\n5. Square Root\n6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

if (choice >= 1 && choice <= 5) { // for options that need input

printf("Enter value: ");

scanf("%lf", &value);

}

switch (choice) {

case 1:

result = sin(value);

printf("Sine(%.2f) = %.4f\n", value, result);

break;

case 2:

result = cos(value);

printf("Cosine(%.2f) = %.4f\n", value, result);

break;

case 3:

if (value > 0) { // log undefined for <= 0

result = log(value);

printf("Log(%.2f) = %.4f\n", value, result);

} else {

printf("Logarithm undefined for <= 0.\n");

}

break;

case 4:

result = exp(value);

printf("e^(%.2f) = %.4f\n", value, result);

break;

case 5:

if (value >= 0) { // sqrt undefined for negative

result = sqrt(value);

printf("Square root(%.2f) = %.4f\n", value, result);

} else {

printf("Square root undefined for negative numbers.\n");

}

break;

case 6:

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

break;

default:

printf("Invalid choice! Try again.\n");

}

} while (choice != 6); // repeat until user chooses Exit

return 0;

}

3. Accept two complex numbers from the user (real part, imaginary part). Write a menu driven program to perform the following operations till the user selects Exit. i. ADD ii. SUBTRACT iii. MULTIPLY iv. EXIT

#include <stdio.h>

struct Complex { // structure to store complex number (a + ib)

double real;

double imag;

};

int main() {

struct Complex c1, c2, result;

int choice;

printf("Enter first complex number (real and imaginary): ");

scanf("%lf %lf", &c1.real, &c1.imag);

printf("Enter second complex number (real and imaginary): ");

scanf("%lf %lf", &c2.real, &c2.imag);

do {

printf("\n--- Complex Number Operations ---\n");

printf("1. Add\n2. Subtract\n3. Multiply\n4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1: // Addition (a+ib) + (c+id) = (a+c) + (b+d)i

result.real = c1.real + c2.real;

result.imag = c1.imag + c2.imag;

printf("Sum = %.2f + %.2fi\n", result.real, result.imag);

break;

case 2: // Subtraction (a+ib) - (c+id) = (a-c) + (b-d)i

result.real = c1.real - c2.real;

result.imag = c1.imag - c2.imag;

printf("Difference = %.2f + %.2fi\n", result.real, result.imag);

break;

case 3: // Multiplication (a+ib)(c+id) = (ac-bd) + (ad+bc)i

result.real = c1.real \* c2.real - c1.imag \* c2.imag;

result.imag = c1.real \* c2.imag + c1.imag \* c2.real;

printf("Product = %.2f + %.2fi\n", result.real, result.imag);

break;

case 4:

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

break;

default:

printf("Invalid choice! Try again.\n");

}

} while (choice != 4); // repeat until Exit selected

return 0;

}