**⭐EXTRA LAB EXERCISES FOR IMPROVING PROGRAMMING LOGIC**

**1. Operators**

* LAB EXERCISE 1: Simple Calculator

Write a C program that acts as a simple calculator. The program should take two numbers and an operator as input from the user and perform the respective operation (addition, subtraction, multiplication, division, or modulus) using operators.

#include<stdio.h>

int main()

{

float n1,n2;

int select;

int a,b;

printf("Enter first number : ");

scanf("%f",&n1);

printf("\nEnter second number : ");

scanf("%f",&n2);

a=n1;

b=n2;

printf("\nEnter what u want to do : ");

printf("\nPress '1' for addition(+)");

printf("\nPress '2' for sub(-)");

printf("\nPress '3' for multi(\*)");

printf("\nPress '4' for division(/)");

printf("\nPress '5' for modules(%%)\n");

scanf("%d",&select);

switch (select)

{

case 1 :

printf("\n %f + %f = %.2f ", n1 , n2, n1+n2);

break;

case 2 :

printf("\n %f - %f = %.2f ", n1 , n2, n1-n2);

break;

case 3 :

printf("\n %f \* %f = %.2f ", n1 , n2, n1\*n2);

break;

case 4 :

printf("\n %f / %f = %.2f ", n1 , n2, n1/n2);

break;

case 5 :

printf("\n %d %% %d = %d ", a , b, a % b );

break;

default:

printf("\nInvalid operator. Please select number from menu");

}

return 0;

}

-----------------------------------------------------------------------

* LAB EXERCISE 2: Check Number Properties

Write a C program that takes an integer from the user and checks the following using different operators:

o Whether the number is even or odd.

o Whether the number is positive, negative, or zero.

o Whether the number is a multiple of both 3 and 5.

#include<stdio.h>

int main()

{

int n;

printf("Enter a number : ");

scanf("%d",&n);

(n % 2 == 0)? printf("this is an even number") : printf("this is an odd number");

(n > 0) ? printf("\n%d is positive", n) :(n < 0) ? printf("\n%d is negative", n) : printf("\n%d is zero", n);

(n%3 == 0 && n%5 == 0) ? printf("\nthe number is a multiple of 3, 5") : printf("\nthe number is NOT a multiple of 3, 5");

return 0;

}

**2. Control Statements**

* LAB EXERCISE 1: Grade Calculator

Write a C program that takes the marks of a student as input and displays the corresponding grade based on the following conditions:

o Marks > 90: Grade A

o Marks > 75 and <= 90: Grade B

o Marks > 50 and <= 75: Grade C

o Marks <= 50: Grade D

• Use if-else or switch statements for the decision-making process.

#include <stdio.h>

int main()

{

int marks;

printf("Enter the students marks : ");

scanf("%d", &marks);

if (marks > 90)

{

printf("\nGrade : A");

}

else if (marks > 75 && marks <= 90)

{

printf("\nGrade : B");

}

else if (marks > 50 && marks <= 75)

{

printf("\nGrade : C");

}

else

{

printf("\nGrade : D");

}

return 0;

}

------------------------------------------------------------------------

* LAB EXERCISE 2: Number Comparison

Write a C program that takes three numbers from the user and determines: o The largest number. o The smallest number.

#include<stdio.h>

int main()

{

int a,b,c,biggest,smallest;

printf("Enter value for 1 num : ");

scanf("%d",&a);

printf("Enter value for 2 num : ");

scanf("%d",&b);

printf("Enter value for 3 num : ");

scanf("%d",&c);

biggest = (a >= b && a >= c) ? a : ( b >= a && b >= c) ? b :c;

smallest = (a <= b && a <= c) ? a : ( b <= a && b <= c) ?b: c;

switch(biggest)

{

case 0 :

break;

default:

printf("\nBiggest number is %d", biggest);

}

switch(smallest)

{

case 0 :

break;

default:

printf("\nSmallest number is %d", smallest);

}

return 0;

}

**3. Loops**

* LAB EXERCISE 1: Prime Number Check

Write a C program that checks whether a given number is a prime number or not using a for loop.

Modify the program to print all prime numbers between 1 and a given number.

#include <stdio.h>

int main()

{

int n, i;

printf("Enter a number : ");

scanf("%d", &n);

if (n == 1)

{

printf("\n%d is not a prime number", n);

return 0;

}

for (i = 2; i < n; i++)

{

if(n % i == 0)

{

printf("\n%d is not a prime number", n);

return 0;

}

}

printf("\n%d is a prime number",n);

return 0;

}

---------------------------modified—------------------------------

#include <stdio.h>

int main()

{

int r, i, j;

printf("Enter the range : ");

scanf("%d", &r);

printf("\nPrime num btwn 1 and %d are : ", r);

for (i = 2; i <= r; i++)

{

for (j = 2; j < i; j++)

{

if (i % j == 0)

{

break;

}

}

if (j == i)

{

printf(" %d ", i);

}

}

printf("\n");

return 0;

}

* LAB EXERCISE 2: Multiplication Table

• Write a C program that takes an integer input from the user and prints its multiplication table using a for loop.

• Challenge: Allow the user to input the range of the multiplication table (e.g., from 1 to N).

#include<stdio.h>

int main()

{

int n,i,r;

printf("Enter the value that u want to multi : ");

scanf("%d",&n);

printf("\nEnter the range : ");

scanf("%d",&r);

for(i=1; i<=r ; i++)

{

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

}

return 0;

}

* LAB EXERCISE 3: Sum of Digits

• Write a C program that takes an integer from the user and calculates the sum of its digits using a while loop.

• Challenge: Extend the program to reverse the digits of the number.

#include<stdio.h>

int main()

{

int num,rnum=0,sum=0;

printf("Enter Any Number : ");

scanf("%d",&num);

while(num>0)

{

int rem = num % 10;

sum+=rem;

rnum = (rnum\*10)+rem;

num=num/10;

}

printf("\nSum of given number : %d",sum);

printf("\nReverse Number is : %d",rnum);

return 0;

}

**4. Arrays**

* LAB EXERCISE 1: Maximum and Minimum in Array

• Write a C program that accepts 10 integers from the user and stores them in an array. The program should then find and print the maximum and minimum values in the array.

• Challenge: Extend the program to sort the array in ascending order.

#include <stdio.h>

int main()

{

int arr[10];

int i, j, temp,max, min;

printf("Enter 10 integers:\n");

for (i = 0; i < 10; i++)

{

scanf("%d", &arr[i]);

}

max = min = arr[0];

for (i = 1; i < 10; i++)

{

if (arr[i] > max)

max = arr[i];

if (arr[i] < min)

min = arr[i];

}

printf("\nMaxi value = %d", max);

printf("\nMini value = %d", min);

for (i = 0; i < 9; i++) //bubble sort

{

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

{

if (arr[i] > arr[j])

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

printf("\nAscending order : ");

for (i = 0; i < 10; i++)

{

printf("%d ", arr[i]);

}

return 0;

}

* LAB EXERCISE 2: Matrix Addition

• Write a C program that accepts two 2x2 matrices from the user and adds them. Display the resultant matrix.

• Challenge: Extend the program to work with 3x3 matrices and matrix multiplication.

#include<stdio.h>

int main()

{

int a[2][2],b[2][2],total[2][2];

printf("Enter 'a' matrix : ");

for(int i=0 ;i<2;i++)

{

for(int j=0;j<2;j++)

{

scanf("%d",&a[i][j]);

}

}

printf("\nEnter 'b' matrix : ");

for(int i=0 ;i<2;i++)

{

for(int j=0;j<2;j++)

{

scanf("%d",&b[i][j]);

}

}

for(int i = 0; i < 2; i++)

{

for(int j = 0; j < 2; j++)

{

total[i][j] = a[i][j] + b[i][j];

}

}

printf("result of a+b : ");

for(int i=0; i<2; i++)

{

for(int j=0;j<2;j++)

{

printf(" %d ",total[i][j]);

}

printf("\n");

}

return 0;

}

-------------------modified version------------------------

#include<stdio.h>

int main()

{

int a[3][3],b[3][3],result[3][3]={0};

int i,j,k;

printf("Enter 1st matrix of 3\*3: ");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

scanf("%d",&a[i][j]);

}

}

printf("Enter 2nd matrix of 3\*3: ");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

scanf("%d",&b[i][j]);

}

}

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

for(k=0;k<3;k++)

{

result[i][j]+= a[i][k]\*b[k][j];

}

}

}

printf("Matrix result :");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

printf(" %d ",result[i][j]);

}

printf("\n");

}

return 0;

}

* LAB EXERCISE 3: Sum of Array Elements

• Write a C program that takes N numbers from the user and stores them in an array. The program should then calculate and display the sum of all array elements.

• Challenge: Modify the program to also find the average of the numbers.

#include <stdio.h>

int main()

{

int n, i;

int arr[100];

int sum = 0;

float avg;

printf("Enter how many values u want to store : ");

scanf("%d", &n);

printf("Enter values : ");

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

{

scanf("%d", &arr[i]);

sum += arr[i];

}

printf("\nSum of given num = %d ", sum);

avg = sum / n;

printf("\nAvg of given array : %f ",avg);

return 0;

}

**5. Functions**

* LAB EXERCISE 1: Fibonacci Sequence

• Write a C program that generates the Fibonacci sequence up to N terms using a recursive function.

• Challenge: Modify the program to calculate the Nth Fibonacci number using both iterative and recursive methods. Compare their efficiency.

**6. Strings**

* LAB EXERCISE 1: String Reversal

• Write a C program that takes a string as input and reverses it using a function.

• Challenge: Write the program without using built-in string handling functions.