AL-KAWTHAR UNIVERSITY

Department of Computer Science

CS 121 L – Programming Fundamentals (PF)

Lab # 04

Objective:

To introduce students to decision-making in C using relational operators (<, <=, >, >=, ==, !=) and conditional statements (if, else). Students will learn to validate input, compare values, and control program flow based on conditions.

Name of Student	**Muhammad Hashir Rafique**
Student ID	**BSCS24101023**
Date of Lab Conducted	
Marks Obtained	
Remarks	
Signature	

LAB 4 - ACTIVITY 1

Relational Operators and Simple Conditions

Objective:

- Demonstrate the use of relational operators to compare values.

Activities:

1. Write a C program to find the maximum value among five user-input variables.
2. Write a C program to find the minimum value among five user-input variables.
3. Dry-run both programs with sample values (e.g., 3, 7, 2, 9, 5).

Example Code (Maximum):

```
#include <stdio.h>

int main() {
    int a, b, c, d, e, max;

    printf("Enter five numbers: ");
    scanf("%d %d %d %d %d", &a, &b, &c, &d, &e);

    max = a;

    if (b > max) max = b;
    if (c > max) max = c;
    if (d > max) max = d;
    if (e > max) max = e;

    printf("Maximum value: %d\n", max);

    return 0;
}
```

C Program to Find Maximum

```c
#include <stdio.h>

int main() {
    int a, b, c, d, e, max;

    printf("Enter five numbers: ");
    scanf("%d %d %d %d %d", &a, &b, &c, &d, &e);

    max = a;
    if (b > max) max = b;
    if (c > max) max = c;
    if (d > max) max = d;
    if (e > max) max = e;

    printf("Maximum value: %d\n", max);
    return 0;
}
```

C Program to Find Minimum

```c
#include <stdio.h>

int main() {
    int a, b, c, d, e, min;

    printf("Enter five numbers: ");
    scanf("%d %d %d %d %d", &a, &b, &c, &d, &e);

    min = a;
    if (b < min) min = b;
    if (c < min) min = c;
    if (d < min) min = d;
    if (e < min) min = e;

    printf("Minimum value: %d\n", min);
    return 0;
}
```

Dry Run

Inputs: `3, 7, 2, 9, 5`

Program 1 (Max):

```
Enter five numbers:
Maximum value: 9
```

Program 2 (Min)

```
Enter five numbers:
Minimum value: 2
```

LAB 4 - ACTIVITY 2

Input Validation with if-else

Objective:

- Validate user input using conditional statements.

Activities:

1. Write a C program that asks for a number between 1 and 9 and prints its multiplication table. If the input is invalid, display an error message.
2. Modify the program to accept a number between 1 and 5 instead.

Example Code:

```
#include <stdio.h>

int main() {
    int num;

    printf("Enter a number (1-9): ");
    scanf("%d", &num);

    if (num >= 1 && num <= 9) {
        for (int i = 1; i <= 10; i++) {
            printf("%d × %d = %d\n", num, i, num * i);
        }
    } else {
        printf("Error: Number out of range!\n");
    }

    return 0;
}
```

C Language Code

```
#include <stdio.h>

int main() {
    int num;

    printf("Enter a number (1 - 5): ");
    scanf("%d", &num);

    if (num >= 1 && num <= 5) {
        for (int i = 1; i <= 10; i++) {
            printf("%d x %d = %d\n", num, i, num * i);
        }} else {
        printf("Error: Number out of range!\n");}

    return 0;}
```

LAB 4 - ACTIVITY 3

Character Case Conversion

Objective:

- Use if-else to manipulate character input.

Activities:

1. Write a C program that takes a character input. If it's uppercase, convert it to lowercase, and vice versa.
2. Display the converted character or an error message for non-alphabetic input.

Example Code:

```
#include <stdio.h>
#include <ctype.h>

int main() {
    char ch;

    printf("Enter a character: ");
    scanf("%c", &ch);

    if (isupper(ch)) {
        printf("Lowercase: %c\n", tolower(ch));
    } else if (islower(ch)) {
        printf("Uppercase: %c\n", toupper(ch));
    } else {
        printf("Error: Not an alphabetic character!\n");
    }

    return 0;
}
```

C Language Code

```
#include <stdio.h>
#include <ctype.h>

int main() {
    char ch;

    printf("Enter a character: ");
    scanf("%c", &ch);

    if (isupper(ch)) {
        printf("Lowercase: %c\n", tolower(ch));
    } else if (islower(ch)) {
        printf("Uppercase: %c\n", toupper(ch));
    } else {
        printf("Error: Not an alphabetic character!\n");
    }

    return 0;
}
```

Dry Run:

```
CASE 1:

Enter a character: D
Lowercase: d

CASE 2:

Enter a character: s
Uppercase: S
```

LAB 4 - ASSIGNMENT

Prepare a document that includes the following:

1. Definitions of relational operators and conditional statements in C.
2. C programs for the following:
   - Check if a user-entered number is even or odd.
   - Ask user to enter a number between 1 and 5, validate the range, and print its table or an error message if the entered number is not within the range.
3. Hand-drawn flowcharts for each program.

---

(1)

Relational Operators in C

Relational operators compare two values. The result is either **true (1)** or **false (0)**.

Operator	Meaning	Example
==	Equal to	a == b
!=	Not equal to	a != b
>	Greater than	a > b
<	Less than	a < b
>=	Greater or equal	a >= b
<=	Less or equal	a <= b

(2)

Program 1:

```
#include <stdio.h>

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    if (num % 2 == 0) {
        printf("%d is even.\n", num);
    } else {
        printf("%d is odd.\n", num);
    }

    return 0;
}
```

FLOWCHART FOR THE PREVIOUS PROGRAM

Program 2:

```
#include <stdio.h>

int main() {
    int num;

    printf("Enter a number (1-5): ");
    scanf("%d", &num);

    if (num >= 1 && num <= 5) {
        printf("Multiplication Table of %d:\n", num);
        for (int i = 1; i <= 10; i++) {
            printf("%d × %d = %d\n", num, i, num * i);
        }
    } else {
        printf("Error: Number not in range (1-5)!\n");
    }

    return 0;
}
```

FLOWCHART FOR THE PREVIOUS PROGRAM