AL-KAWTHAR UNIVERSITY

Department of Computer Science

CS 121 L – Programming Fundamentals (PF)

Lab # 06

Objective:

To introduce students to the concept of arrays in C programming. Students will learn how to declare, initialize, and manipulate arrays for storing and processing multiple values efficiently.

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

LAB 6 - ACTIVITY 1

Visualizing Arrays with ASCII Art

Objective:

- Demonstrate the concept of arrays by visually representing them using ASCII characters..

Activities:

1. Declare an array of size 5 to represent "gray boxes" using ASCII character.
2. Check the ASCII chart for the gray shaded box.
3. Initialize the array with placeholder values.
4. Print the array elements to display the boxes in a horizontal line.
5. Modify the program to print the boxes using the ASCII code from ASCII chart.

Example Code:

```
#include <stdio.h>

int main() {
    char boxes[5] = {'X', 'X', 'X', 'X', 'X'};

    printf("Array of boxes:\n");
    for (int i = 0; i < 5; i++) {
        printf("[%c] ", boxes[i]);
    }

    return 0;
}
```

C Language code:

```
#include <stdio.h>

int main() {
    // ASCII value 177 = shaded gray box
    char allBoxes[5] = {177, 177, 177, 177, 177};

    printf("Array of gray boxes:\n");
    for (int i = 0; i < 5; i++) {
        printf("[%c] ", allBoxes[i]); }

    return 0;}
```

Output:

```
Array of gray boxes:
[▒] [▒] [▒] [▒] [▒]
```

LAB 6 - ACTIVITY 2

Concatenating First and Last Name

Objective:

- Use character arrays to store and concatenate a user's first and last name.

Activities:

1. Declare two character arrays: firstName and lastName.
2. Prompt the user to input their first and last name.
3. Concatenate the two arrays into a third array (fullName) and print the result.
    - Enter your first name: Hashir
    - Enter your last name: Rafique
    - Full Name: Hashir Rafique

Example Code:

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

int main() {
    char firstName[20], lastName[20], fullName[40];

    printf("Enter your first name: ");
    scanf("%s", firstName);

    printf("Enter your last name: ");
    scanf("%s", lastName);

    strcpy(fullName, firstName);
    strcat(fullName, " ");
    strcat(fullName, lastName);

    printf("Full Name: %s\n", fullName);

    return 0;
}
```

C Language code:

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

int main() {
    char firstName[20], lastName[20], fullName[40];

    printf("Enter your first name: ");
    scanf("%s", firstName);

    printf("Enter your last name: ");
    scanf("%s", lastName);

    strcpy(fullName, firstName);
    strcat(fullName, " ");
    strcat(fullName, lastName);

    printf("Your full name is: %s\n", fullName);

    return 0;}
```

Output:

```
Enter your first name: Hashir
Enter your last name: Rafique
Your full name is: Hashir Rafique
```

LAB 6 - ACTIVITY 3

Finding Maximum and Minimum in an Array

Objective:

- Use an array to store five numbers and determine the maximum and minimum values.

Activities:

1. Declare an integer array of size 5.
2. Prompt the user to input five numbers and store them in the array.
3. Traverse the array to find and print the maximum and minimum values.
    - Enter 5 numbers: 3 9 2 7 5
    - Maximum: 9
    - Minimum: 2
4. Dry-run the program and explain the loop iterations.

Example Code:

```
#include <stdio.h>

int main() {
    int numbers[5], i, max, min;

    printf("Enter 5 numbers: ");
    for (i = 0; i < 5; i++) {
        scanf("%d", &numbers[i]);
    }

    max = min = numbers[0];
    for (i = 1; i < 5; i++) {
        if (numbers[i] > max)
            max = numbers[i];
        if (numbers[i] < min)
            min = numbers[i];
    }

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

    return 0;}
```

C Language Code

```c
#include <stdio.h>

int main() {
    int numbers[5], i, maximum, min;

    printf("Enter 5 numbers: ");
    for (i = 0; i < 5; i++) {
        scanf("%d", &numbers[i]);
    }

    maximum = min = numbers[0];

    for (i = 1; i < 5; i++) {
        if (numbers[i] > maximum)
            maximum = numbers[i];
        if (numbers[i] < min)
            min = numbers[i];
    }

    printf("Maximum: %d\n", maximum);
    printf("Minimum: %d\n", min);

    return 0;
}
```

Dry Run:

```
Enter 5 numbers: 3 9 2 7 5
Maximum: 9
Minimum: 2
```

LAB 6 - ASSIGNMENT

Prepare a document that includes the following:

1. Programs in C language:
    - Modify Activity 2 to concatenate without using string functions
    - Extend Activity 3 to sort the array in:
        - Ascending order
        - Descending order
2. Flowcharts: Hand-drawn for each program.

---

(1)

Manual Concatenation:

```
#include <stdio.h>

int main() {
    char firstName[20], lName[20], fName[40];
    int i = 0, j = 0;

    printf("Enter your first name: ");
    scanf("%s", firstName);

    printf("Enter your last name: ");
    scanf("%s", lName);

    while (firstName[i] != '\0') {
        fName[i] = firstName[i];
        i++;
}

    fName[i] = ' ';
    i++;

    while (lName[j] != '\0') {
        fName[i] = lName[j];
        i++;
        j++;}

    fName[i] = '\0';

    printf("Full Name: %s\n", fName);

    return 0;
}
```

FLOWCHART OF PREVIOUS PROGRAM:

Ascending and descending order:

```
#include <stdio.h>
int main() {
    int numbers[5], i, j, temp;
    printf("Enter 5 numbers: ");
    for (i = 0; i < 5; i++) {
        scanf("%d", &nums[i]);}
    // Ascending Order
    for (i = 0; i < 4; i++) {
        for (j = i + 1; j < 5; j++) {
            if (nums[i] > nums[j]) {
                temp = nums[i];
                nums[i] = nums[j];
                nums[j] = temp; } }}
    printf("Ascending order: ");
    for (i = 0; i < 5; i++) {
        printf("%d ", nums[i]);}
    // Descending Order
    for (i = 0; i < 4; i++) {
        for (j = i + 1; j < 5; j++) {
            if (nums[i] < nums[j]) {
                temp = nums[i];
                nums[i] = nums[j];
                nums[j] = temp; }  } }
    printf("\nDescending order: ");
    for (i = 0; i < 5; i++) {
        printf("%d ", nums[i]);
    }
    return 0;}
```

FLOWCHART OF PREVIOUS PROGRAM: