SDM COLLEGE OF ENGINEERING AND TECHNOLOGY

Dhavalagiri, Dharwad-580002, Karnataka State, India.

Email: cse.sdmcet@gmail.com

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A Report

on

Course work/Assignment for CTA

COURSE CODE: 22UCSC501 **COURSE TITLE: Database Management System**
SEMESTER: 5 DIVISION: A **COURSE TEACHER: U.P.Kulkarni**

[Academic Year- 2023-24]

Submitted By

Name: Ms. Ananya U Gaonkar **USN: 2SD22CS012**

Table of Contents

A1: Write a C program to study all file operations related SYSTEM CALLS supported by UNIX OS and C libraries for file operations.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
int main() {
// if file does not exist in directory, source.txt is created
// Open a source file for reading and writing
int source_fd = open("source.txt", O_RDWR | O_CREAT, 0644);
// O_RDWR    allows both reading and writing
if (source_fd == -1) {
perror("Failed to open source.txt");
exit(1); }
// Writing into the file source.txt
int sz = write(source_fd, "AnanyaGaonkar", strlen("AnanyaGaonkar"));
printf("%d\n", sz);
// Move the file offset to the beginning of the file
lseek(source_fd, 0, SEEK_SET); // Reset the offset to the start for reading
// Create or open a destination file for writing
int dest_fd = open("destination.txt", O_WRONLY | O_CREAT | O_TRUNC, 0644);
if (dest_fd == -1) {
perror("Failed to open destination.txt");
```

```
close(source_fd); // Close the source file
exit(1);
}
// Read from the source file and write to the destination file
char buffer[4096]; // A buffer to hold data
ssize_t nread;

while ((nread = read(source_fd, buffer, sizeof(buffer))) > 0) {
if (write(dest_fd, buffer, nread) != nread) {
perror("Write error");
break;
}
}
// Check if there was an error during reading
if (nread < 0) {
perror("Read error");
}
// Close both files
close(source_fd);
close(dest_fd);

return 0;
}
```

A2 .Write a C program to demonstrate indexing and associated operations.

```
#include <stdio.h>
// Function to display the array
void display(int arr[], int n) {
printf("Array elements: ");
for (int i = 0; i < n; i++) {
printf("%d ", arr[i]);
}
printf("\n");
}
// Function to insert an element at a given index
void insert(int arr[], int *n, int element, int index) {
if (index >= 0 && index <= *n) {
for (int i = *n; i > index; i--) {
arr[i] = arr[i - 1];
}
arr[index] = element;
(*n)++;
printf("Inserted %d at index %d\n", element, index);
} else {
printf("Invalid index\n");
}
}
```

```
// Function to delete an element at a given index
void delete(int arr[], int *n, int index) {
if (index >= 0 && index < *n) {
printf("Deleted %d from index %d\n", arr[index], index);
for (int i = index; i < *n - 1; i++) {
arr[i] = arr[i + 1];
}
(*n)--;
} else {
printf("Invalid index\n");
}
}
// Function to update an element at a given index
void update(int arr[], int n, int element, int index) {
if (index >= 0 && index < n) {
printf("Updated index %d from %d to %d\n", index, arr[index], element);
arr[index] = element;
} else {
printf("Invalid index\n");
}
}
int main() {
int arr[100];
```

```
int n = 0; // Number of elements in the array
int choice, element, index;

do {
printf("\nMenu:\n");
printf("1. Insert\n");
printf("2. Delete\n");
printf("3. Update\n");
printf("4. Display\n");
printf("5. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);

switch (choice) {
case 1:
printf("Enter element to insert: ");
scanf("%d", &element);
printf("Enter index to insert at: ");
scanf("%d", &index);
insert(arr, &n, element, index);
break;
case 2:
printf("Enter index to delete from: ");
scanf("%d", &index);
delete(arr, &n, index);
```

```
break;
case 3:
printf("Enter new element to update: ");
scanf("%d", &element);
printf("Enter index to update at: ");
scanf("%d", &index);
update(arr, n, element, index);
break;
case 4:
display(arr, n);
break;
case 5:
printf("Exiting...\n");
break;
default:
printf("Invalid choice!\n");
}
} while (choice != 5);
return 0;
}
```

A3: Write a java program to access a given excel file with known format

```
import java.io.File;
import java.io.FileInputStream;
import java.util.Iterator;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;

public class ExcelFileReading {

public static void main(String[] args) {
try {
FileInputStream file = new FileInputStream(new File("input.xlsx"));
XSSFWorkbook workbook = new XSSFWorkbook(file);
XSSFSheet sheet = workbook.getSheetAt(0);
Iterator<Row> rowIterator = sheet.iterator();

while (rowIterator.hasNext()) {
Row row = rowIterator.next();
Iterator<Cell> cellIterator = row.cellIterator();

while (cellIterator.hasNext()) {
```

```
Cell cell = cellIterator.next();


switch (cell.getCellType()) {
case NUMERIC:
System.out.print(cell.getNumericCellValue() + "\t");
break;
case STRING:
System.out.print(cell.getStringCellValue() + "\t");
break;
default:
System.out.print("Unknown type\t");
break;
}
}
System.out.println("");
}
file.close();
workbook.close();
} catch (Exception e) {
e.printStackTrace();
}
}
}
```