OSLAB1 & 2

1. Create the following directories with one command.

   OSSPRING2025/OSLAB -> OSLAB1

```
student@student-A110SU:~$ mkdir -p OSSPRING2025/OSLAB/OSLAB1
```

2. Create a group name 'OperatingSystemLab1'
3. Create a user account 'OSUser1' and 'OSUser2' and add it to the group 'OperatingSystemLab1'. Login in to that user using terminal.

```
student@student-A110SU:~$ sudo groupadd OperatingSystemLab1
[sudo] password for student:
student@student-A110SU:~$ sudo useradd -m OSUser1 -g OperatingSystemLab1
student@student-A110SU:~$ sudo useradd -m OSUser2 -g OperatingSystemLab1
student@student-A110SU:~$ sudo passwd OSUser1
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
student@student-A110SU:~$ sudo passwd OSUser2
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simpli
tematic
Retype new password:
passwd: password updated successfully
student@student-A110SU:~$ su - OSUser1
Password:
```

4. Create a file 'file1.txt' and write "LinuxOperating system".
5. Create another file 'file2.txt'.
6. Copy the content of 'file1.txt' into 'file2.txt'.

```
student@student-A110SU:~$ touch "file1.txt"
student@student-A110SU:~$ gedit "file1.txt"
student@student-A110SU:~$ ls -l "file1.txt"
-rw-rw-r-- 1 student student 23 08:41 7   فروری file1.txt
student@student-A110SU:~$ ls
Desktop    Downloads  file2.txt  OSSPRING2025  Public  Templates
Documents  file1.txt  Music      Pictures      snap    Videos
student@student-A110SU:~$ touch "file2.txt"
student@student-A110SU:~$ cp "file1.txt" "file2.txt"
student@student-A110SU:~$ cat "file2.txt"
LinuxOperating system

student@student-A110SU:~$ cat "file1.txt"
LinuxOperating system
```

7. On one line, use the "cd" command to first go to your home directory then to the rollnumber subdirectory. [Ans: cd/home; cd rollnumber]

```
student@student-A110SU:~$ mkdir 23K0842
student@student-A110SU:~$ cd ~; cd 23K0842
student@student-A110SU:~/23K0842$ ls
student@student-A110SU:~/23K0842$ pwd
/home/student/23K0842
```

8. Explain the difference between the 'mv' and 'cp' commands.

mv (move):

- The mv command is used to move or rename files or directories.
- It removes the original file or directory after moving it to the new location.
- Example: Moving a file from one directory to another.

cp (copy):

- The cp command is used to copy files or directories.
- It creates a duplicate of the source file or directory in the specified destination without modifying the original file.
- Example: Copying a file to another location.

9. How would you move a file named "doc.txt" to a directory named "documents"?

mv doc.txt documents/

10. Write a C++ program that uses the <cmath> library to calculate the square root of a number. Compile and run the program.

```
student@student-A110SU:~$ touch "sqrt_program.cpp"
student@student-A110SU:~$ gedit "sqrt_program.cpp"
```

```cpp
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    double num=4.0;

    if (num < 0) {
        cout<<"Error: Negative number entered."<<endl;
        return 1;
    }

    double result = sqrt(num);
    cout<<"The square root of "<<num<<" is "<<result<<endl;

    return 0;
}
```

```
student@student-A110SU:~$ g++ "sqrt_program.cpp" -o out
student@student-A110SU:~$ ./out
The square root of 4 is 2
```

11. Write a C++ program that initializes an array of integers and finds the sum of its elements. Compile and run the program.

```cpp
#include <iostream>
using namespace std;
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int sum = 0;

    for (int i = 0; i < 5; i++) {
        sum += arr[i];
    }

    cout<< "The sum of the array elements is: "<<sum <<endl;

    return 0;
}
```

```
student@student-A110SU:~$ touch "sum.cpp"
student@student-A110SU:~$ gedit "sum.cpp"
student@student-A110SU:~$ g++ "sum.cpp" -o out
student@student-A110SU:~$ ./out
The sum of the array elements is: 15
```

12. Write a C++ program that takes a string as a command line argument and checks whether it is a palindrome or not.

```
k200281kainat@k200281kainat-VirtualBox:~$ touch "palindrome.cpp"
k200281kainat@k200281kainat-VirtualBox:~$ gedit "palindrome.cpp"
k200281kainat@k200281kainat-VirtualBox:~$ g++ "palindrome.cpp" -o out
k200281kainat@k200281kainat-VirtualBox:~$ ./out 2 madam
Palindromek200281kainat@k200281kainat-VirtualBox:~$
```

```cpp
#include<iostream>
#include<cstring>
using namespace std;
bool ispalindrome(const char *str)
{
int len=strlen(str);
for(int i=0;i<len/2;i++)
{
if(str[i] != str[len-i-1])
{
return false;
}
}
return true;
}
int main(int argc,char *argv[])
{
if(ispalindrome(argv[1]))
{
cout<<"Palindrome";
}
else
{
cout<<"Not a Palindrome"<<endl;
}
return 0;
}
```

13. Write a C++ program that acts as a simple calculator. It should take three command line arguments: two numbers and an operation (+, -, *, /) and print the result.

```
k200281kainat@k200281kainat-VirtualBox:~$ touch "calc.cpp"
k200281kainat@k200281kainat-VirtualBox:~$ gedit "calc.cpp"
```

```
k200281kainat@k200281kainat-VirtualBox:~$ g++ "calc.cpp" -o out
```

```
#include<iostream>
#include<cstdlib>
using namespace std;
int main(int argc,char *argv[])
{
double num1 = atof(argv[1]);
char op = argv[2][0];
double num2 = atof(argv[3]);
double result;
switch(op)
{
case '+':
result= num1+num2;
break;
case '-':
result= num1-num2;
break;
case '*':
result= num1*num2;
break;
case '/':
if(num2==0)
{cout<<"Error....Division by zero"<<endl;
}
result = num1/num2;
break;
default:
cout<<"Invalid..."<<endl;
return 1;
}
cout<<"Result: "<<result<<endl;
return 0;
}
```

```
k200281kainat@k200281kainat-VirtualBox:~$ ./out 10 "+" 5
Result: 15
k200281kainat@k200281kainat-VirtualBox:~$ ./out 5 "*" 5
Result: 25
k200281kainat@k200281kainat-VirtualBox:~$ ./out 4 "/" 4
Result: 1
k200281kainat@k200281kainat-VirtualBox:~$ ./out 4 "-" 2
Result: 2
```

14. Your task is to develop a simple Student Management System in C that allows users to add a student, display all students, and search for a student by ID. Organize your code into five files: main.c (handles the main menu), add_student.c (adds student records), display_students.c (displays all students), search_student.c (searches for a student by ID), and student.h (defines the Student structure with fields like id and name, and declares function prototypes).
    - You must write a Makefile to compile all .c files into a single executable named student_mgmt, with a clean target to remove the executable. Compile the program using make, run it with ./student_mgmt, and clean up using make clean.
    - The program should display a menu with options to add, display, search for students, and exit. It should loop until the user chooses to exit. Handle invalid inputs appropriately.
    - **Submission**: Zip all source files (.c, .h, Makefile) as StudentManagement_<YourName> with terminal screenshots showing successful compilation and execution.

```c
#ifndef STUDENT_H
#define STUDENT_H

#define MAX_NAME_LENGTH 50
#define MAX_STUDENTS 100

typedef struct
{
int id;
char name[MAX_NAME_LENGTH];
} Student;

extern Student students[MAX_STUDENTS];
extern int student_count;
void add_student();
void display_students();
void search_student();

#endif
```

Student.c file

```c
#include "student.h"
#include<stdio.h>

Student students[MAX_STUDENTS];
int student_count = 0;
void add_student()
{
if(student_count >=MAX_STUDENTS)
{
printf("Students list is full.\n");
return;
}
printf("Enter Student ID: ");
scanf("%d",&students[student_count].id);
printf("Enter Student Name: ");
scanf(" %s",students[student_count].name);
student_count++;
printf("Student added successfully\n");
}
```

search_student.c file

```c
#include "student.h"
#include<stdio.h>

void search_student()
{
int id;
printf("Enter id for student to search: ");
scanf("%d", &id);

for (int i=0;i<student_count;i++)
{
if(students[i].id == id)
{
printf("ID: %d, Name: %s\n",students[i].id, students[i].name);
return;
}
}
printf("Student not found\n");
}
```

display_students.c file

```c
#include "student.h"
#include<stdio.h>

void display_students()
{
if(student_count == 0)
{
printf("No student to display");
return;
}
for (int i=0;i<student_count;i++)
{
printf("ID: %d, Name: %s\n",students[i].id, students[i].name);
}
}
```

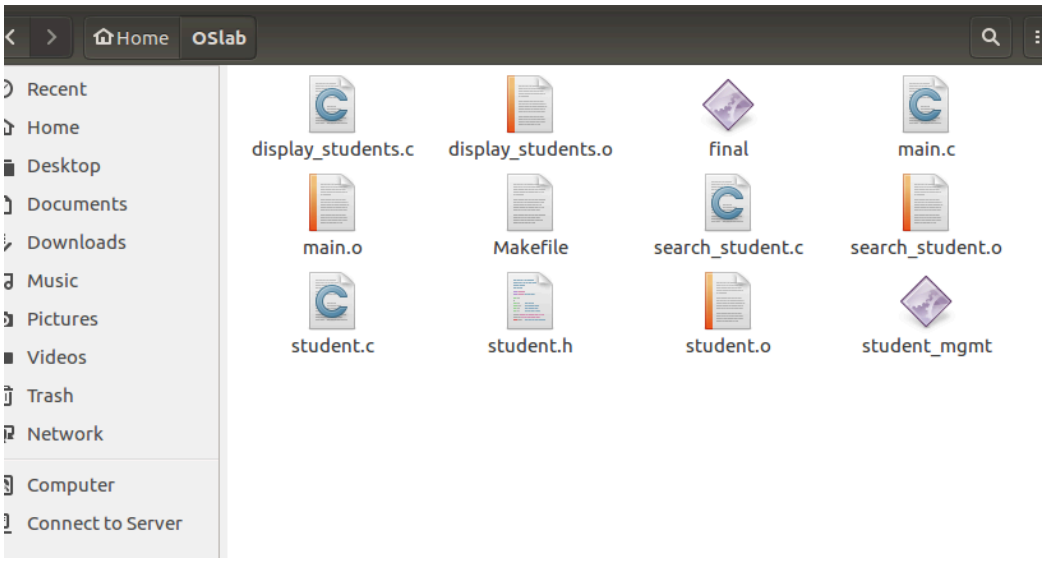
main.c file

```c
#include "student.h"
#include<stdio.h>

int main()
{
int choice;
while(1)
{
printf("Student Management System \n");
printf("1. Add Student\n");
printf("2. Display Student\n");
printf("3. Search Student\n");
printf("4. Exit\n");
printf("Enter your choice: ");
scanf("%d",&choice);

switch(choice)
{
case 1:
add_student();
break;
case 2:
display_students();
break;
case 3:
search_student();
break;
case 4:
printf("Exiting");
return 0;
default:
printf("Invalid choice\n");
}
}
return 0;
}
```

Makefile

```
all: student_mgmt

student_mgmt: main.o student.o display_students.o search_student.o
	gcc main.o student.o display_students.o search_student.o -o student_mgmt

main.o: main.c
	gcc -c main.c

student.o: student.c
	gcc -c student.c

display_students.o: display_students.c
	gcc -c display_students.c

search_student.o: search_student.c
	gcc -c search_student.c

clean:
	rm -rf *o student_mgmt
```

```
k200281kainat@k200281kainat-VirtualBox:~/OSlab$ make all
gcc -c main.c
gcc -c student.c
gcc -c display_students.c
gcc -c search_student.c
gcc main.o student.o display_students.o search_student.o -o student_mgmt
k200281kainat@k200281kainat-VirtualBox:~/OSlab$ ./student_mgmt
Student Management System
1. Add Student
2. Display Student
3. Search Student
4. Exit
Enter your choice: 1
Enter Student ID: 12
Enter Student Name: kinza
Student added successfully
Student Management System
1. Add Student
2. Display Student
3. Search Student
4. Exit
Enter your choice: 2
ID: 12, Name: kinza
Student Management System
1. Add Student
2. Display Student
3. Search Student
4. Exit
Enter your choice: 3
Enter id for student to search: 12
ID: 12, Name: kinza
Student Management System
1. Add Student
2. Display Student
3. Search Student
4. Exit
Enter your choice: 4
Exitingk200281kainat@k200281kainat-VirtualBox:~/OSlab$
```



```
k200281kainat@k200281kainat-VirtualBox:~/OSlab$ make clean
rm -rf *o student_mgmt
k200281kainat@k200281kainat-VirtualBox:~/OSlab$
```