

# <<Lift Management>>

## 21CSS101J – PROGRAMMING FOR PROBLEM SOLVING

### Mini Project Report

*Submitted by*

**Jevil Mansata [Reg. No.: RA2211003011862]**

**B.Tech. CSE - << Core >>**

**Pranjal Babel [Reg. No.: RA2211030010312]**

**B.Tech. CSE - << Cyber Security >>**



**SCHOOL OF COMPUTING  
COLLEGE OF ENGINEERING AND TECHNOLOGY  
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

**S.R.M. NAGAR, KATTANKULATHUR – 603 203**

**KANCHEEPURAM DISTRICT**

**December 2022**

## **TABLE OF CONTENTS**

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
1	Problem Statement	2-3
2	Methodology / Procedure	5 & 8-10
3	Coding (C or Python)	6-7
4	Results	11
5	Conclusion	12

DECEMBER 10TH 2022

# LIFT MANAGEMENT

PPS MINI PROJECT  
DOCUMENTATION

Submitted by:

Jevil Mansata (RA2211003011862)

Pranjal Babel (RA2211030010312)

# 02

## ABOUT LIFT MANAGEMENT PROJECT

Purpose of this program is to suggest students the fastest lift route (as per The SRM University Building Lift format) to reach the floor they wish to go. The program calculate the estimate time to reach the floor by each lift respectively and suggests the fastest route possible.

## BENEFITS

- This program enables students to make more efficient approach while choosing the right lift queue, in order to reach the destination floor, the fastest.
- Helps to reach early classes on time.
- Designed as per current SRM-UB lift format. it is practically applicable.
- Helps university management, to diversify students, in order to make system organized.

## 3

## SRM UB LIFT FORMAT

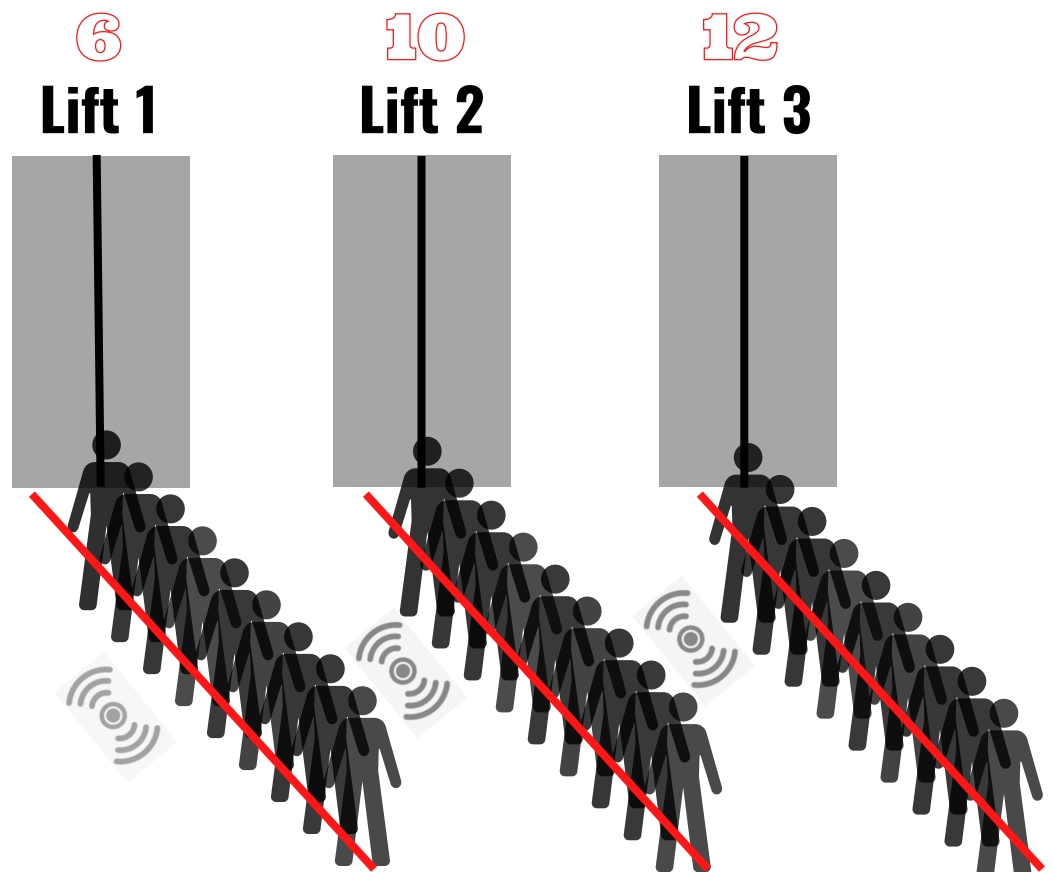
The current SRM Lift format are as follows:

- There are 3 lifts in total for boys.
- Rush hours ( for example 7:30/8:00 am )

lift 1: 6th floor lift 2: 10th floor lift 3: 12th floor

In rush hours the floors are distributed in 3 lifts.

Lift just goes to specific floor allotted.



## 04

## FACTORS/VARIABLE

**These are factors program takes into considers and their short form codes used in program:**

1. PQ1: people in queue 1
2. PQ2: people in queue 2
3. PQ3: people in queue 3
4. TTS: time taken to cross one floor by stairs
5. TTL: time taken to cross one floor by lift
6. MF1: maximum floor lift 1
7. MF2: maximum floor lift 2
8. MF3: maximum floor lift 3
9. EF1 : Extra floors covered by stairs - lift1
10. EF2 : Extra floors covered by stairs - lift2
11. EF3 : Extra floors covered by stairs - lift3
12. F : Destination Floor entered by user
13. TT1 : Time Taken by lift 1 to reach MF1
13. TT2 : Time Taken by lift 2 to reach MF2
13. TT3 : Time Taken by lift 3 to reach MF3
14. FST1: Final stairs time lift 1
15. FST2: Final stairs time lift 2
15. FST3: Final stairs time lift 3
16. FLT1: Final lift time lift 1
17. FLT2: Final lift time lift 2
18. FLT3: Final lift time lift 3
19. EST1: Estimated time by lift 1
20. EST2: Estimated time by lift 2
21. EST3: Estimated time by lift 3
22. LTL1: Lift turns left in lift 1
23. LTL2: Lift turns left in lift 2
24. LTL3: Lift turns left in lift 3
25. QWT1: queue waiting time in lift 1
26. QWT2: queue waiting time in lift 2
27. QWT3: queue waiting time in lift 3

# 05

## HOW PROGRAM WORKS:

- First the program takes input from the sensors near queues, to get number of people standing in each lifts queue respectively.
- Then program will take input from the user to get the floor destination.
- Program will use following factors into consideration:
  - 1) Number of people in line
  - 2) Total time in Queue
  - 3) Total time in lift
  - 4) Total time to cover extra floors by stairs
- Program will calculate the following factors above and provide an estimate time to reach the destination floor in each lift respectively.
- Program will compare the estimate time in each lift and provide user the fastest lift route possible.

```

#include<stdio.h>
void main()
{
int pq1,pq2,pq3,mf1=6,mf2=10,mf3=12,ef1,ef2,ef3,f,mp=15,ltl1,ltl2,ltl3,q1,q2,q3;
float tts=15,ttl=8,est1,est2,est3,tt1=32,tt2=64,tt3=80,qwt1,qwt2,qwt3,fst1,fst2,fst3,flt1,flt2,flt3;
printf("Enter the number of people in each lift queue:\n");
printf("Lift 1: ");
scanf("%d",&pq1);
printf("Lift 2: ");
scanf("%d",&pq2);
printf("Lift 3: ");
scanf("%d",&pq3);

q1=pq1/15;
q2=pq2/15;
q3=pq3/15;

printf("Turn in Lift 1: %d\n",q1);
printf("Turn in Lift 2: %d\n",q2);
printf("Turn in Lift 3: %d\n",q3);

qwt1=(q1)*(2)*(tt1);
qwt2=(q2)*(2)*(tt2);
qwt3=(q3)*(2)*(tt3);

printf("Waiting time for Lift 1: %.2f\n",qwt1);
printf("Waiting time for Lift 2: %.2f\n",qwt2);
printf("Waiting time for Lift 3: %.2f\n",qwt3);

printf("Enter the floor: ");
scanf("%d",&f);

if(f<mf1){
ef1=mf1-f;
ef2=mf2-f;
ef3=mf3-f;
}

if(f>mf1 && f<mf2){
ef1=f-mf1;
ef2=mf2-f;
ef3=mf3-f;
}

if(f>mf2 && f<mf3){
ef1=f-mf1;
ef2=f-mf2;
ef3=mf3-f;
}

if(f>mf3 && f<=15){
ef1=f-mf1;
ef2=f-mf2;
ef3=f-mf3;
}

```



```

if(f==mf1){
ef1=0;
ef2=mf2-f;
ef3=mf3-f;
}

if(f==mf2){
ef1=f-mf1;
ef2=0;
ef3=mf3-f;
}

if(f==mf3){
ef1=f-mf1;
ef2=f-mf2;
ef3=0;
}

printf("Extra floors in Queue 1: %d\n",ef1);
printf("Extra floors in Queue 2: %d\n",ef2);
printf("Extra floors in Queue 3: %d\n",ef3);

fst1=ef1*tts;
fst2=ef2*tts;
fst3=ef3*tts;

    printf("Time taken by stairs by Queue 1: %.2f\n",fst1);
    printf("Time taken by stairs by Queue 2: %.2f\n",fst2);
    printf("Time taken by stairs by Queue 3: %.2f\n",fst3);

est1=fst1+qwt1+tt1;
est2=fst2+qwt2+tt2;
est3=fst3+qwt3+tt3;

printf("Estimate time by Lift 1: %.2f\n",est1);
printf("Estimate time by Lift 2: %.2f\n",est2);
printf("Estimate time by Lift 3: %.2f\n",est3);

if(est1<est2 && est1<est3){
printf("Fastest option: Lift 1");
}

else if(est2<est3){
printf("Fastest option: Lift 2");
}

else{
printf("Fastest option: Lift 3");
}

return 0;
}

```

## 08

## PART 1

```
#include<stdio.h>
void main()
{
    int pq1,pq2,pq3,mf1=6,mf2=10,mf3=12,ef1,ef2,ef3,f,mp=15,ltl1,ltl2,ltl3,q1,q2,q3;
    float tts=15,ttl=8,est1,est2,est3,ttl=32,tt2=64,tt3=80,qwt1,qwt2,qwt3,fst1,fst2,fst3,flt1,flt2,flt3;
    printf("Enter the number of people in each lift queue:\n");
    printf("Lift 1: ");
    scanf("%d",&pq1);
    printf("Lift 2: ");
    scanf("%d",&pq2);
    printf("Lift 3: ");
    scanf("%d",&pq3);
```

- VARIABLE DECLARATION
- TAKING INPUTS FROM THE SENSOR, TO ACHIEVE PEOPLE STANDING QUEUE.

## PART 2

```
q1=pq1/15;
q2=pq2/15;
q3=pq3/15;

printf("Turn in Lift 1: %d\n",q1);
printf("Turn in Lift 2: %d\n",q2);
printf("Turn in Lift 3: %d\n",q3);

qwt1=(q1)*(2)*(tt1);
qwt2=(q2)*(2)*(tt2);
qwt3=(q3)*(2)*(tt3);

printf("Waiting time for Lift 1: %.2f\n",qwt1);
printf("Waiting time for Lift 2: %.2f\n",qwt2);
printf("Waiting time for Lift 3: %.2f\n",qwt3);
```

- PERSON IN QUEUE GETS DIVIDED TO THE MAXIMUM LIFT CAPACITY (IE. 15) AND IT GETS STORED INTO VARIABLE Q, WHICH INDICATES LIFT TURNS LEFT.
- THEN QUEUE WAITING TIME IS CALCULATED BY MULTIPLYING LIFT TURNS LEFT, AND TIME TAKEN BY LIFT TO REACH ITS MAXIMUM FLOOR AND COME BACK.

## 09

## PART 3

```
printf("Enter the floor: ");
scanf("%d",&f);
```

- FLOOR INPUT IS TAKEN FROM THE USER

```
if(f<mf1){
    ef1=mf1-f;
    ef2=mf2-f;
    ef3=mf3-f;
}
```

```
if(f==mf1){
    ef1=0;
    ef2=mf2-f;
    ef3=mf3-f;
}
```

```
if(f>mf1 && f<mf2){
    ef1=f-mf1;
    ef2=mf2-f;
    ef3=mf3-f;
}
```

```
if(f==mf2){
    ef1=f-mf1;
    ef2=0;
    ef3=mf3-f;
}
```

```
if(f>mf2 && f<mf3){
    ef1=f-mf1;
    ef2=f-mf2;
    ef3=mf3-f;
}
```

```
if(f==mf3){
    ef1=f-mf1;
    ef2=f-mf2;
    ef3=0;
}
```

```
if(f>mf3 && f<=15){
    ef1=f-mf1;
    ef2=f-mf2;
    ef3=f-mf3;
}
```

```
printf("Extra floors in Queue 1: %d\n",ef1);
printf("Extra floors in Queue 2: %d\n",ef2);
printf("Extra floors in Queue 3: %d\n",ef3);
```

- EXTRA FLOORS THAT HAS TO COVERED BY STAIRS ARE BEING CALCULATED IN THIS STEP.
- BY USING DIFFERENT IF STATEMENTS AND RANGES, EXTRA FLOORS ARE CALCULATED BY SUBTRACTING FLOOR BY MAXIMUM FLOOR/ MAXIMUM FLOOR BY FLOOR. ACCORDING TO THE CASE.

## PART 4

```
fst1=ef1*tts;
fst2=ef2*tts;
fst3=ef3*tts;
```

- FINAL STAIRS TIME IS CALCULATED BY MULTIPLYING EXTRA FLOORS TO TIME TAKEN TO CROSS ONE FLOOR BY STAIRS.

```
printf("Time taken by stairs by Queue 1: %.2f\n",fst1);
printf("Time taken by stairs by Queue 2: %.2f\n",fst2);
printf("Time taken by stairs by Queue 3: %.2f\n",fst3);
```

```
est1=fst1+qwt1+tt1;
est2=fst2+qwt2+tt2;
est3=fst3+qwt3+tt3;
```

- ESTIMATE TIME IS CALCULATED BY ADDING FINAL STAIRS TIME ,QUEUE WAITING TIME AND TIME TAKEN BY LIFT TO REACH MAXIMUM FLOOR.

```
printf("Estimate time by Lift 1: %.2f\n",est1);
printf("Estimate time by Lift 2: %.2f\n",est2);
printf("Estimate time by Lift 3: %.2f\n",est3);
```

10

## PART 5

```
if(est1<est2 && est1<est3){  
    printf("Fastest option: Lift 1");  
}  
  
else if(est2<est3){  
    printf("Fastest option: Lift 2");  
}  
  
else{  
    printf("Fastest option: Lift 3");  
}  
  
return 0;
```

- AT LAST, ALL THREE LIFTS ESTIMATE TIME IS COMPARED USING IF ELSE STATEMENT, TO ACHIEVE THE LOWEST ESTIMATE TIME. AND IT PRINTS TO DECLARE THE LIFT WITH LOWEST ESTIMATE TIME AS THE FASTEST OPTION.

## 11

## OUTPUT/RESULTS

PROGRAM FIRST TAKES INPUT FROM THE ADMIN/SENSORS TO GET PEOPLE IN QUEUE OF EACH LIFT. THEN PROGRAM CALCULATES AND DISPLAYS WAITING TIME IN EACH LIFT. THEN PROGRAM TAKES INPUT FROM THE USER TO GET THEIR DESTINATION FLOOR. THEN PROGRAM CALCULATES AND DISPLAYS, EXTRA FLOORS, TIME TAKEN BY STAIRS AND EVENTUALLY ESTIMATED TIME BY EACH LIFT. PROGRAM FINALLY PRINTS THE FASTEST OPTION.

## Output

```
/tmp/uEW0bRTX0J.o
```

```
Enter the number of people in each lift queue:
```

```
Lift 1: |
```

## Output

```
/tmp/uEW0bRTX0J.o
```

```
Enter the number of people in each lift queue:
```

```
Lift 1: 45
```

```
Lift 2: 43
```

```
Lift 3: 38
```

```
Turn in Lift 1: 3
```

```
Turn in Lift 2: 2
```

```
Turn in Lift 3: 2
```

```
Waiting time for Lift 1: 192.00
```

```
Waiting time for Lift 2: 256.00
```

```
Waiting time for Lift 3: 320.00
```

```
Enter the floor: 11
```

```
Extra floors in Queue 1: 5
```

```
Extra floors in Queue 2: 1
```

```
Extra floors in Queue 3: 1
```

```
Time taken by stairs by Queue 1: 75.00
```

```
Time taken by stairs by Queue 2: 15.00
```

```
Time taken by stairs by Queue 3: 15.00
```

```
Estimate time by Lift 1: 299.00
```

```
Estimate time by Lift 2: 335.00
```

```
Estimate time by Lift 3: 415.00
```

```
Fastest option: Lift 1|
```

## Output

```
/tmp/uEW0bRTX0J.o
```

```
Enter the number of people in each lift queue:
```

```
Lift 1: 45
```

```
Lift 2: 43
```

```
Lift 3: 38
```

```
Turn in Lift 1: 3
```

```
Turn in Lift 2: 2
```

```
Turn in Lift 3: 2
```

```
Waiting time for Lift 1: 192.00
```

```
Waiting time for Lift 2: 256.00
```

```
Waiting time for Lift 3: 320.00
```

```
Enter the floor: |
```

## CONCLUSION:

*"This program will boost the efficiency of current lift model. It will also help students choose the best route available to their destination floor in rush hours."*

This program will revolutionize the way students use the current lift model. Students just have to enter their destination floor in the program and program will simply calculate the best route possible in that current moment. Students will be able to reach their classes on time in rush hours. it will also help university management to make the current model, more organized and efficient. It will to diversify students to different mediums, rather than gathering in one resource.