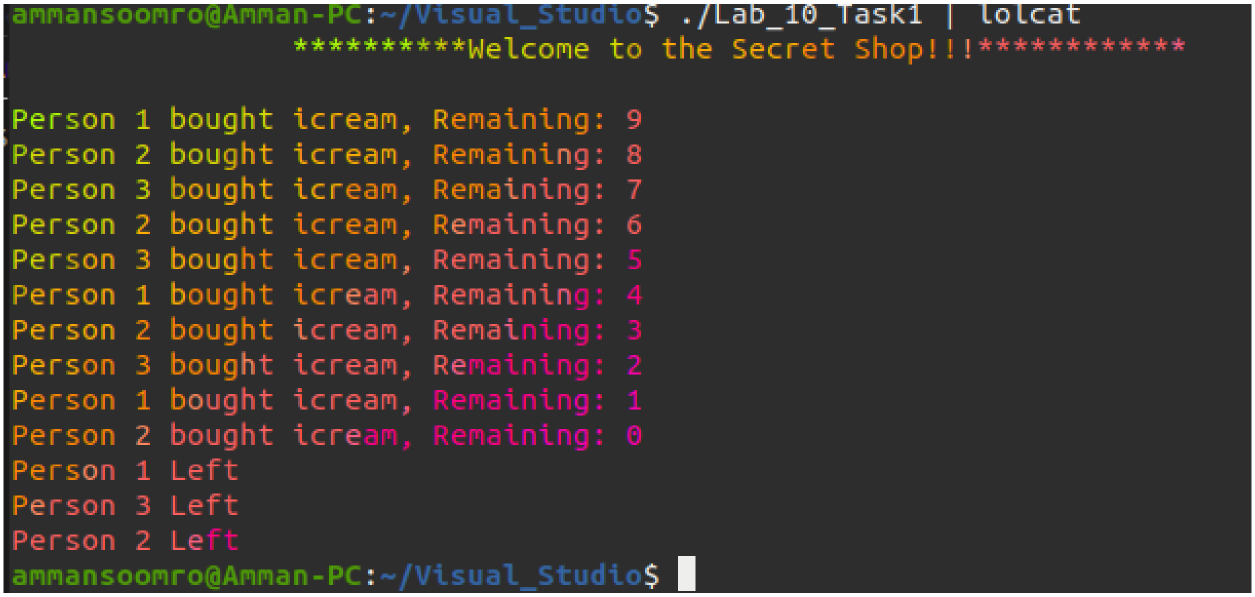
K191048
Amman Soomro

TASK 1 (A)

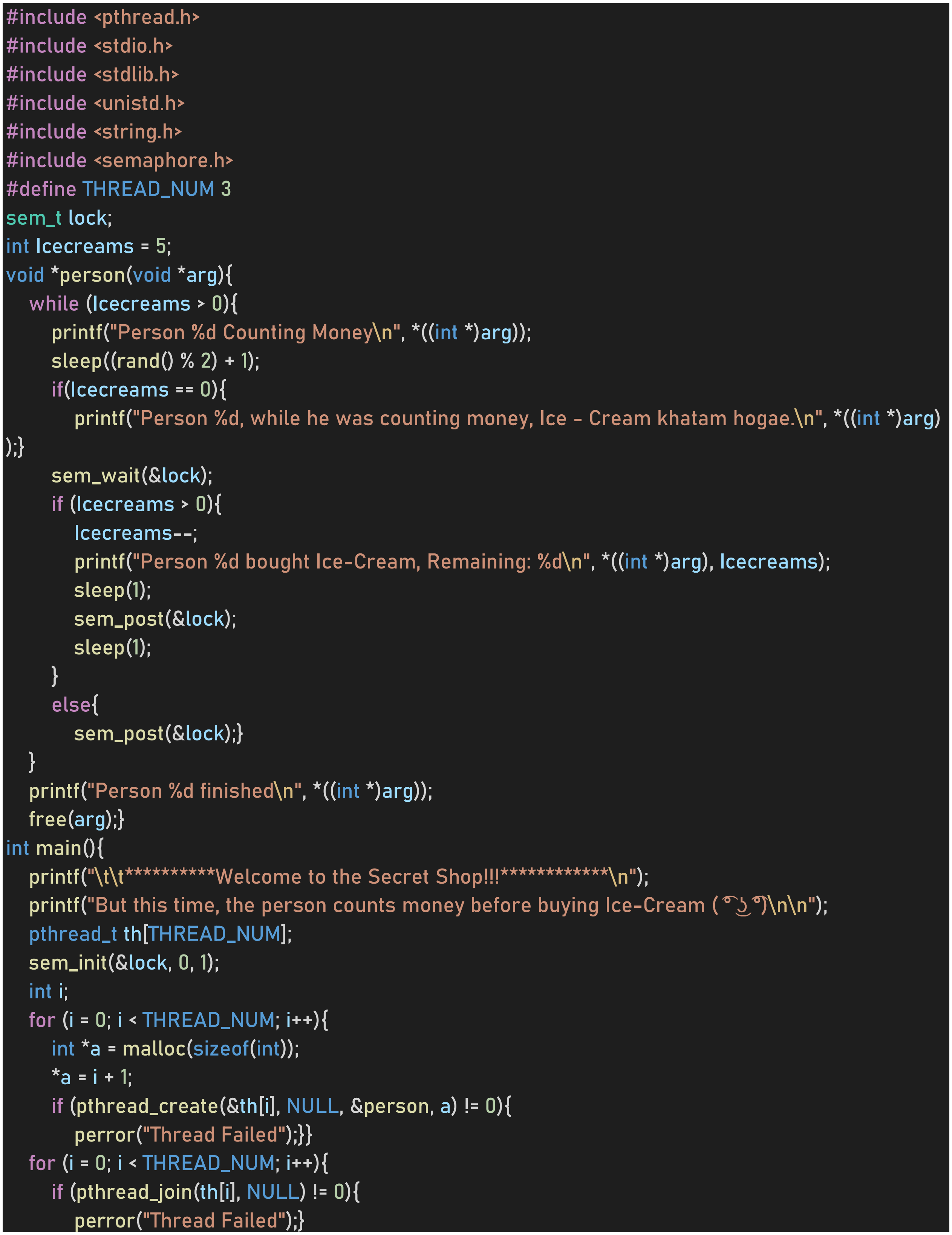
```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <semaphore.h>
#define THREAD_NUM 3
sem_t lock;
int Icecreams = 10;
void *person(void *arg){
    while (Icecreams != 0){
        sem_wait(&lock);
        Icecreams--;
        printf("Person %d bought IceCream, Remaining: %d\n", *((int *)arg), Icecreams);
        sem_post(&lock);
        sleep(1);}
    printf("Person %d Left\n", *((int *)arg));
    free(arg);
}
int main()
{
    printf("\t\t**********Welcome to the Secret Shop!!!************\n\n");
    pthread_t th[THREAD_NUM];
    sem_init(&lock, 0, 1);
    int i;
    for (i = 0; i < THREAD_NUM; i++)
    {
        int *a = malloc(sizeof(int));
        *a = i + 1;
        if (pthread_create(&th[i], NULL, &person, a) != 0){
            perror("Thread Failed.");}}
    for (i = 0; i < THREAD_NUM; i++)
    {
        if (pthread_join(th[i], NULL) != 0)
        {
            perror("Thread Failed.");
        }
    }
    sem_destroy(&lock);
    return 0;
}
```



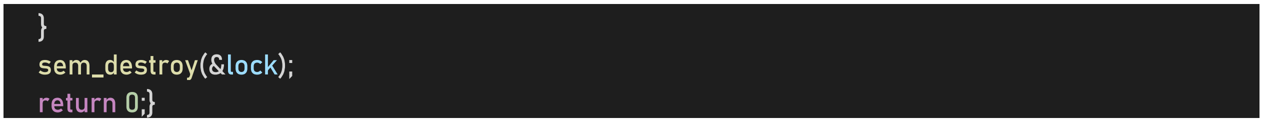
```
ammansoomro@Amman-PC:~/Visual_Studio$ ./Lab_10_Task1 | lolcat
                    **********Welcome to the Secret Shop!!!************

Person 1 bought icream, Remaining: 9
Person 2 bought icream, Remaining: 8
Person 3 bought icream, Remaining: 7
Person 2 bought icream, Remaining: 6
Person 3 bought icream, Remaining: 5
Person 1 bought icream, Remaining: 4
Person 2 bought icream, Remaining: 3
Person 3 bought icream, Remaining: 2
Person 1 bought icream, Remaining: 1
Person 2 bought icream, Remaining: 0
Person 1 Left
Person 3 Left
Person 2 Left
ammansoomro@Amman-PC:~/Visual_Studio$ 
```

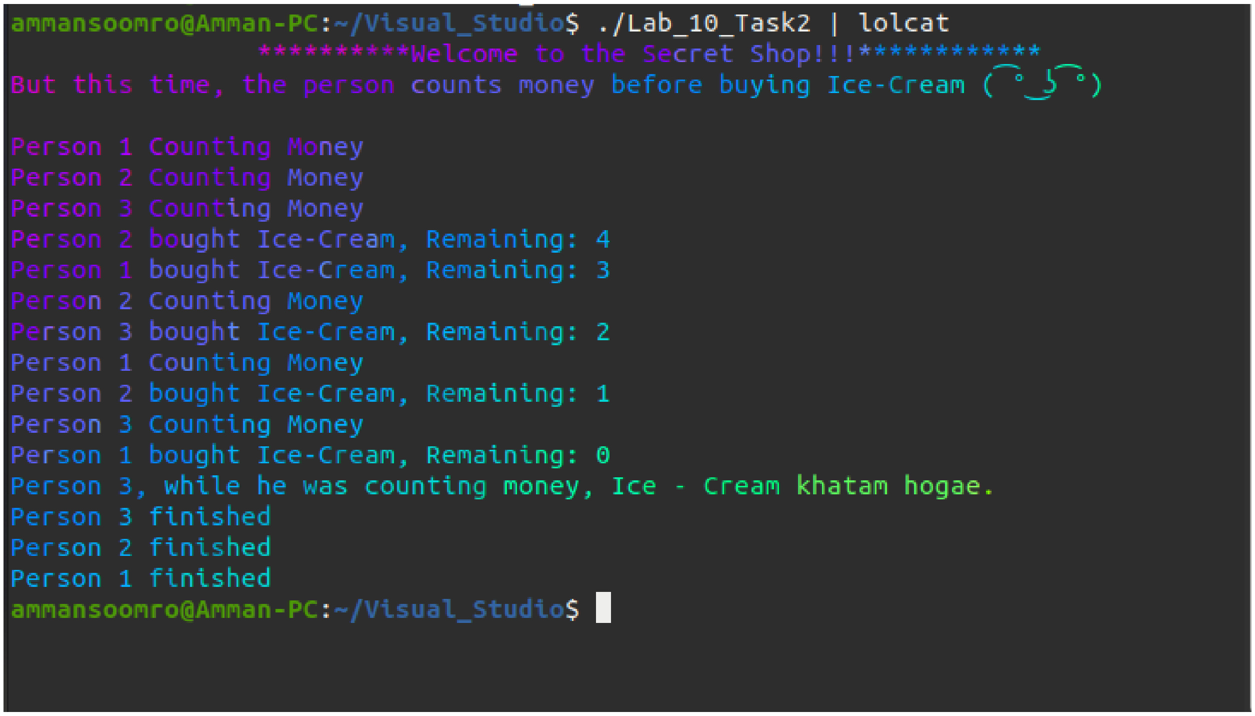
TASK 1 (B)



```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <semaphore.h>
#define THREAD_NUM 3
sem_t lock;
int Icecreams = 5;
void *person(void *arg){
    while (Icecreams > 0){
        printf("Person %d Counting Money\n", *((int *)arg));
        sleep((rand() % 2) + 1);
        if(Icecreams == 0){
            printf("Person %d, while he was counting money, Ice - Cream khatam hogae.\n", *((int *)arg)
);}
        sem_wait(&lock);
        if (Icecreams > 0){
            Icecreams--;
            printf("Person %d bought Ice-Cream, Remaining: %d\n", *((int *)arg), Icecreams);
            sleep(1);
            sem_post(&lock);
            sleep(1);
        }
        else{
            sem_post(&lock);}
    }
    printf("Person %d finished\n", *((int *)arg));
    free(arg);}
int main(){
    printf("\t\t**********Welcome to the Secret Shop!!!************\n");
    printf("But this time, the person counts money before buying Ice-Cream ( ͡° ͜ʖ ͡°)\n\n");
    pthread_t th[THREAD_NUM];
    sem_init(&lock, 0, 1);
    int i;
    for (i = 0; i < THREAD_NUM; i++){
        int *a = malloc(sizeof(int));
        *a = i + 1;
        if (pthread_create(&th[i], NULL, &person, a) != 0){
            perror("Thread Failed");}}
    for (i = 0; i < THREAD_NUM; i++){
        if (pthread_join(th[i], NULL) != 0){
            perror("Thread Failed");}
```



```
  }
  sem_destroy(&lock);
  return 0;}
```



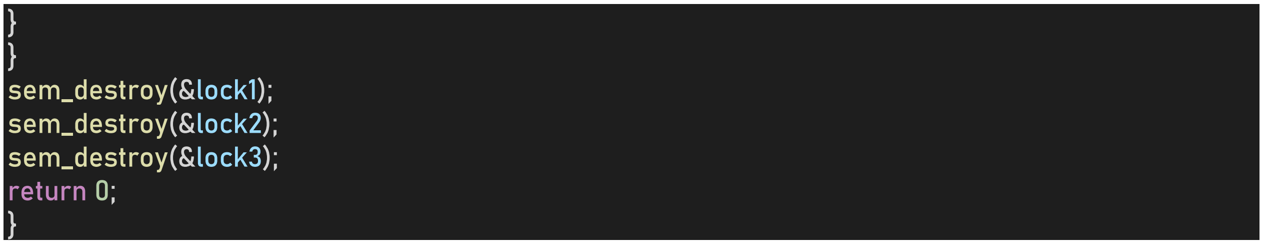
```
ammansoomro@Amman-PC:~/Visual_Studio$ ./Lab_10_Task2 | lolcat
                ***********Welcome to the Secret Shop!!!************
But this time, the person counts money before buying Ice-Cream ( ͡° ͜ʖ ͡°)

Person 1 Counting Money
Person 2 Counting Money
Person 3 Counting Money
Person 2 bought Ice-Cream, Remaining: 4
Person 1 bought Ice-Cream, Remaining: 3
Person 2 Counting Money
Person 3 bought Ice-Cream, Remaining: 2
Person 1 Counting Money
Person 2 bought Ice-Cream, Remaining: 1
Person 3 Counting Money
Person 1 bought Ice-Cream, Remaining: 0
Person 3, while he was counting money, Ice - Cream khatam hogae.
Person 3 finished
Person 2 finished
Person 1 finished
ammansoomro@Amman-PC:~/Visual_Studio$ 
```

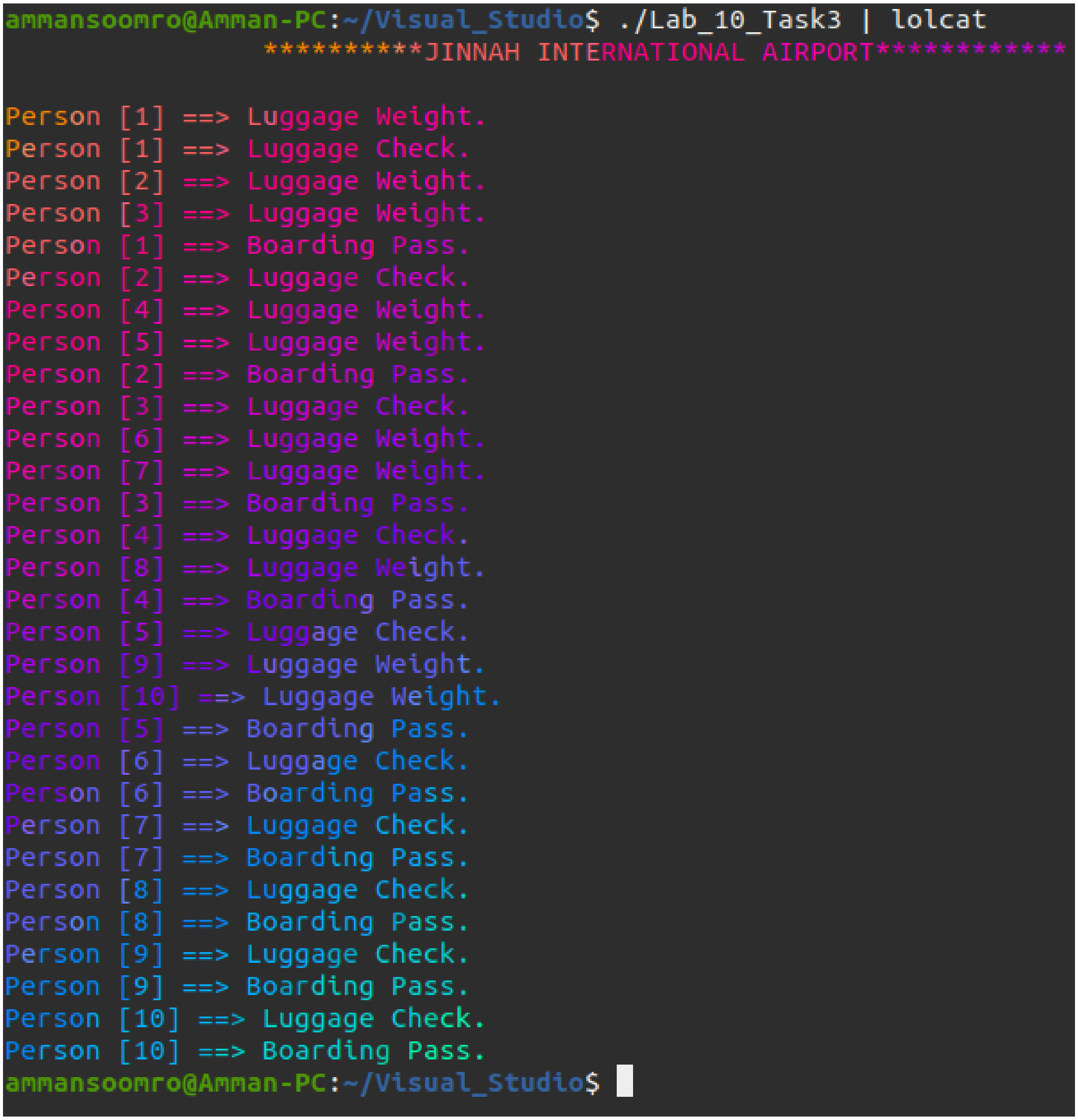
TASK (2)

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <semaphore.h>
#define THREAD_NUM 10
sem_t lock1;
sem_t lock2;
sem_t lock3;
void *person(void *arg)
{
sem_wait(&lock1);
printf("Person [%d] ==> Luggage Weight.\n", *((int *)arg));
sleep(4);
sem_post(&lock1);
sem_wait(&lock2);
printf("Person [%d] ==> Luggage Check.\n", *((int *)arg));
sleep(7);
sem_post(&lock2);
sem_wait(&lock3);
printf("Person [%d] ==> Boarding Pass.\n", *((int *)arg));
sleep(3);
sem_post(&lock3);
free(arg);
}
```

```
int main()
{
printf("\t\t**********JINNAH INTERNATIONAL AIRPORT************\n\n");
pthread_t th[THREAD_NUM];
sem_init(&lock1, 0, 1);
sem_init(&lock2, 0, 1);
sem_init(&lock3, 0, 1);
int i;
for (i = 0; i < THREAD_NUM; i++)
{
int *a = malloc(sizeof(int));
*a = i + 1;
if (pthread_create(&th[i], NULL, &person, a) != 0)
{
perror("Thread Failed.");
}
}
for (i = 0; i < THREAD_NUM; i++)
{
if (pthread_join(th[i], NULL) != 0)
{
perror("Thread Failed.");
```



```
}
}
sem_destroy(&lock1);
sem_destroy(&lock2);
sem_destroy(&lock3);
return 0;
}
```



```
ammansoomro@Amman-PC:~/Visual_Studio$ ./Lab_10_Task3 | lolcat
                    **********JINNAH INTERNATIONAL AIRPORT************

Person [1] ==> Luggage Weight.
Person [1] ==> Luggage Check.
Person [2] ==> Luggage Weight.
Person [3] ==> Luggage Weight.
Person [1] ==> Boarding Pass.
Person [2] ==> Luggage Check.
Person [4] ==> Luggage Weight.
Person [5] ==> Luggage Weight.
Person [2] ==> Boarding Pass.
Person [3] ==> Luggage Check.
Person [6] ==> Luggage Weight.
Person [7] ==> Luggage Weight.
Person [3] ==> Boarding Pass.
Person [4] ==> Luggage Check.
Person [8] ==> Luggage Weight.
Person [4] ==> Boarding Pass.
Person [5] ==> Luggage Check.
Person [9] ==> Luggage Weight.
Person [10] ==> Luggage Weight.
Person [5] ==> Boarding Pass.
Person [6] ==> Luggage Check.
Person [6] ==> Boarding Pass.
Person [7] ==> Luggage Check.
Person [7] ==> Boarding Pass.
Person [8] ==> Luggage Check.
Person [8] ==> Boarding Pass.
Person [9] ==> Luggage Check.
Person [9] ==> Boarding Pass.
Person [10] ==> Luggage Check.
Person [10] ==> Boarding Pass.
ammansoomro@Amman-PC:~/Visual_Studio$ 
```