**Name : Ibadullah Shaikh**

**Roll No : 19K-0259**

**Section : BCS-4H**

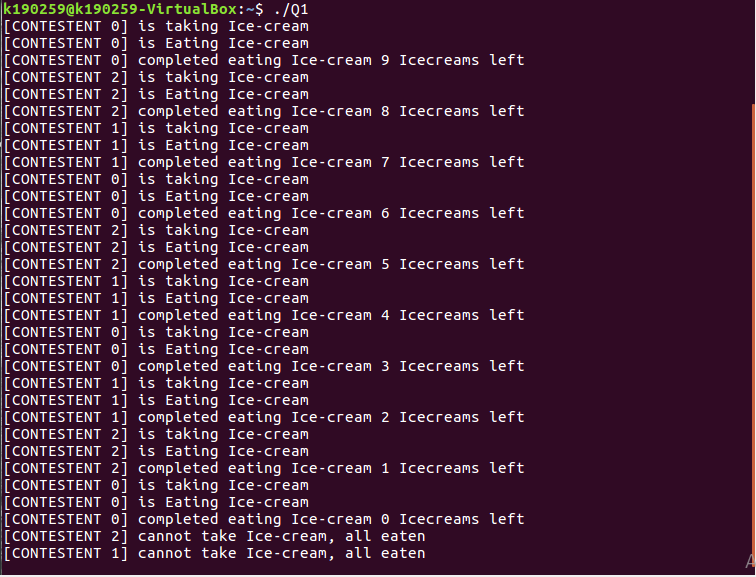
**Assignment No # 07**

**Q1 (a)**

**Code:**

#include<stdio.h>  
#include<semaphore.h>  
#include<pthread.h>  
#include<sys/types.h>  
#include<unistd.h>  
int icecreams=10;  
int numpeople=3;  
sem\_t lock;  
void \* eaticecream(void \*args)  
{  
int x;  
int \*ptr=(int\*)args;  
int val=\*ptr;  
sem\_wait(&lock);  
if(icecreams > 0)  
{  
printf("[CONTESTENT %d] is taking Ice-cream\n",val);  
x=icecreams;  
printf("[CONTESTENT %d] is Eating Ice-cream\n",val);  
x--;  
icecreams=x;  
printf("[CONTESTENT %d] completed eating Ice-cream %d Icecreams left\n",val,icecreams);  
}  
else  
{  
printf("[CONTESTENT %d] cannot take Ice-cream, all eaten\n",val);  
}  
sem\_post(&lock);  
return NULL;  
}  
int main()  
{  
int p[numpeople];  
sem\_init(&lock,0,1);  
int contestants[numpeople];  
pthread\_t tcont[5];  
for(int i=0;i<numpeople;i++)  
{  
contestants[i]=i;  
}  
int i=0;  
while(icecreams>0)  
{  
for(i=0;i<numpeople;i++)  
{  
pthread\_create(&tcont[i],NULL,eaticecream, (void\*)&contestants[i]);  
}  
pthread\_join(tcont[0],NULL);  
pthread\_join(tcont[1],NULL);  
pthread\_join(tcont[2],NULL);  
sem\_destroy(&lock);  
}  
return 0;  
}

**Screenshot:**

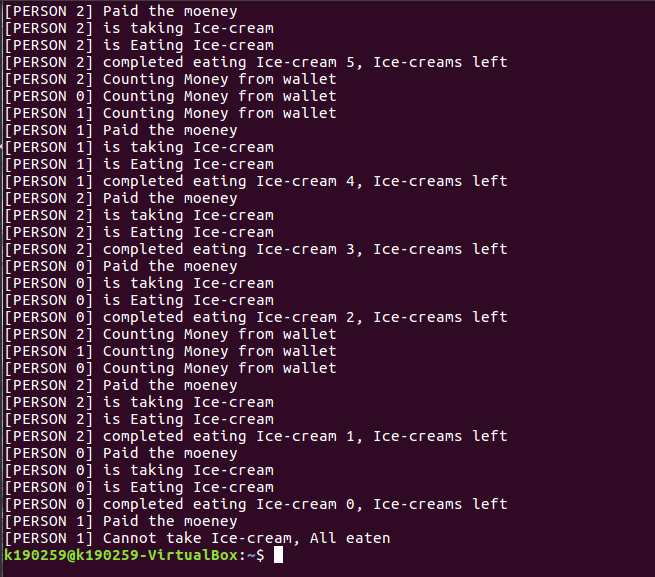


**Q1 (b)**

**Code:**

#include<stdio.h>  
#include<semaphore.h>  
#include<pthread.h>  
#include<sys/types.h>  
#include<unistd.h>  
int icecreams=8;  
int numpeople=3;  
sem\_t lock;  
sem\_t ilock;  
void \*eaticecream(void \*args)  
{  
int x;  
int \*ptr=(int\*)args;  
int val=\*ptr;  
printf("[PERSON %d] Counting Money from wallet\n",val);  
sleep(2);  
printf("[PERSON %d] Paid the moeney\n",val);  
sem\_wait(&lock);  
if(icecreams>0)  
{  
printf("[PERSON %d] is taking Ice-cream\n",val);  
x = icecreams;  
printf("[PERSON %d] is Eating Ice-cream\n",val);  
x--;  
icecreams=x;  
printf("[PERSON %d] completed eating Ice-cream %d, Ice-creams left\n",val,icecreams);  
}  
else  
{  
printf("[PERSON %d] Cannot take Ice-cream, All eaten\n", val);  
}  
sem\_post(&lock);  
return NULL;  
}  
int main()  
{  
int p[numpeople];  
sem\_init(&lock , 0 , 1);  
int contestants[numpeople];  
pthread\_t tcont[5];  
for(int i=0;i<numpeople;i++)  
{  
contestants[i]=i;  
}  
int i =0;  
while(icecreams>0)  
{  
for(i=0;i<numpeople;i++)  
{  
pthread\_create(&tcont[i],NULL,eaticecream, (void\*)&contestants[i]);  
}  
pthread\_join(tcont[0],NULL);  
pthread\_join(tcont[1],NULL);  
pthread\_join(tcont[2],NULL);  
sem\_destroy(&lock);  
}  
return 0;  
}

**Screenshot:**



**Q2**

**Code:**

#include<stdio.h>  
#include<semaphore.h>  
#include<pthread.h>  
#include<sys/types.h>  
#include<unistd.h>  
sem\_t locks[3];  
void\* travel(void \*args)  
{  
int ind=0;  
int \*ptr=(int\*)args;  
int val=\*ptr;  
printf("[Customer %d] is going to weight luggage\n",val);  
sem\_wait(&locks[ind]);  
printf("[Customer %d] Weighting his/her luggage\n",val);  
sleep(4);  
printf("[Customer %d] just Completed weighting luggage\n",val);  
sem\_post(&locks[ind]);  
ind++;  
printf("[Customer %d] is Going to Security check\n",val);  
sem\_wait(&locks[ind]);  
printf("[Customer %d] Entered Security Check\n",val);  
sleep(7);  
printf("[Customer %d] Completed Security Check\n",val);  
sem\_post(&locks[ind]);  
ind++;  
printf("[Customer %d] Going to Boarding pass\n",val);  
sem\_wait(&locks[ind]);  
printf("[Customer %d] is Getting Boarding pass\n",val);  
sleep(7);  
printf("[Customer %d] Got Boarding Pass\n",val);  
sem\_post(&locks[ind]);  
return NULL;  
}  
int main()  
{  
for(int i=0;i<3;i++)  
{  
sem\_init(&locks[i] , 0 , 1);  
}  
pthread\_t cust[10];  
int customers[10];  
for(int i=0;i<10;i++)  
{  
customers[i]=i;  
}  
for(int i=0;i<10;i++)  
{  
pthread\_create(&cust[i],NULL,travel,(void\*)&customers[i]);  
}  
for(int i=0;i<10;i++)  
{  
pthread\_join(cust[i],NULL);  
}  
for(int i=0;i<3;i++)  
{  
sem\_destroy(&locks[i]);  
}  
return 0;  
}

**Screenshot:**

