Operating Systems

UE20CS254

Name: Naman Choudhary	SRN: PES2UG20CS209	Section: D

Week

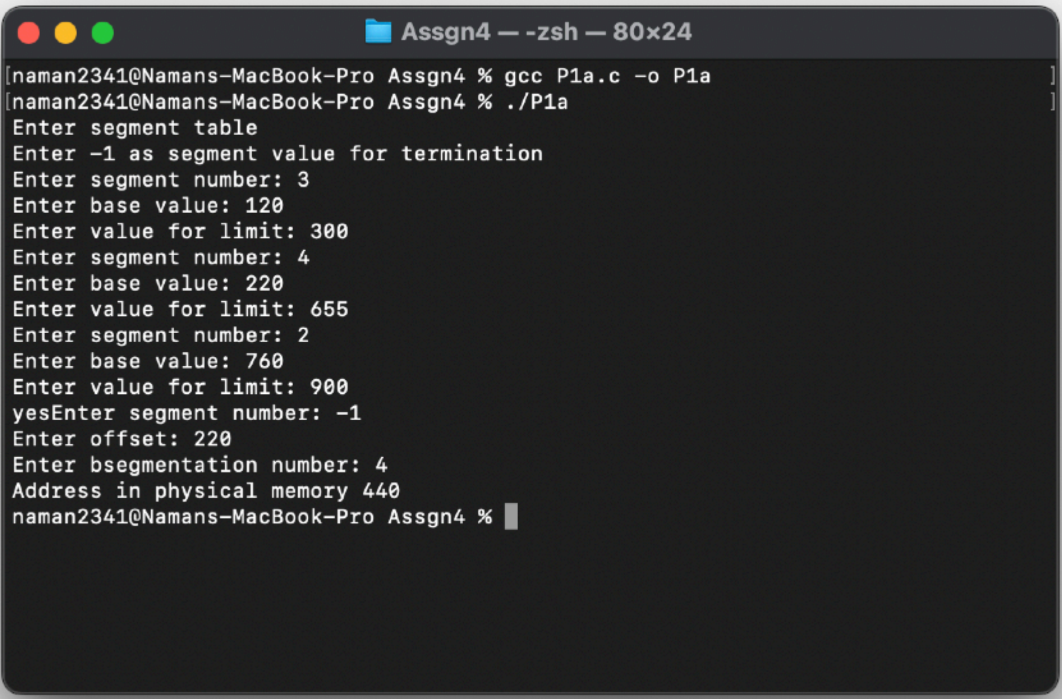
Program Number	1 a
Program Qn	Write a C program to simulate Segmentation Take as input: 1. Segment number 2. Base address 3. Segment limit

Source Code

```
#include<stdio.h>
#include<stdlib.h>
struct list {
    int seg;
    int base;
    int limit;
    struct list *next;
} *p;
void insert(struct list *q,int base,int limit,int seg) {
    if(p==NULL)
    {
        p=malloc(sizeof(struct list));
        p->limit=limit;
        p->base=base;
        p->seg=seg;
        p->next=NULL;
    }
```

```c
else
    {
        while(q->next!=NULL)
        {
q=q->next;
            printf("yes");
        }
        q->next=malloc(sizeof(struct list));
        q->next ->limit=limit;
        q->next ->base=base;
        q->next ->seg=seg;
        q->next ->next=NULL;
    }
}
int find(struct list *q,int seg)
{
    while(q->seg!=seg)
    {
q=q->next; }
    return q->limit;
}
int search(struct list *q,int seg)
{
    while(q->seg!=seg)
    {
q=q->next; }
    return q->base;
}
int main() {
    p=NULL;
    int seg,offset,limit,base,c,s,physical;
    printf("Enter segment table\n");
    printf("Enter -1 as segment value for 
termination\n");
    do
    {
        printf("Enter segment number: ");
        scanf("%d",&seg);
        if(seg!=-1)
        {
        printf("Enter base value: ");
        scanf("%d",&base);
        printf("Enter value for limit: ");
        scanf("%d",&limit);
        insert(p,base,limit,seg);
} }
    while(seg!=-1);
    printf("Enter offset: ");
    scanf("%d",&offset);
    printf("Enter bsegmentation number: ");
    scanf("%d",&seg);
    c=find(p,seg);
    s=search(p,seg);
    if(offset<c)
    {
        physical=s+offset;
        printf("Address in physical memory 
%d\n",physical);
    }
else
    {
        printf("error");
} }
```

Output Screenshot



```
naman2341@Namans-MacBook-Pro Assgn4 % gcc P1a.c -o P1a
naman2341@Namans-MacBook-Pro Assgn4 % ./P1a
Enter segment table
Enter -1 as segment value for termination
Enter segment number: 3
Enter base value: 120
Enter value for limit: 300
Enter segment number: 4
Enter base value: 220
Enter value for limit: 655
Enter segment number: 2
Enter base value: 760
Enter value for limit: 900
yesEnter segment number: -1
Enter offset: 220
Enter bsegmentation number: 4
Address in physical memory 440
naman2341@Namans-MacBook-Pro Assgn4 %
```

Program Number	1 b
Program Qn	C program for LRU replacement algorithm implementation

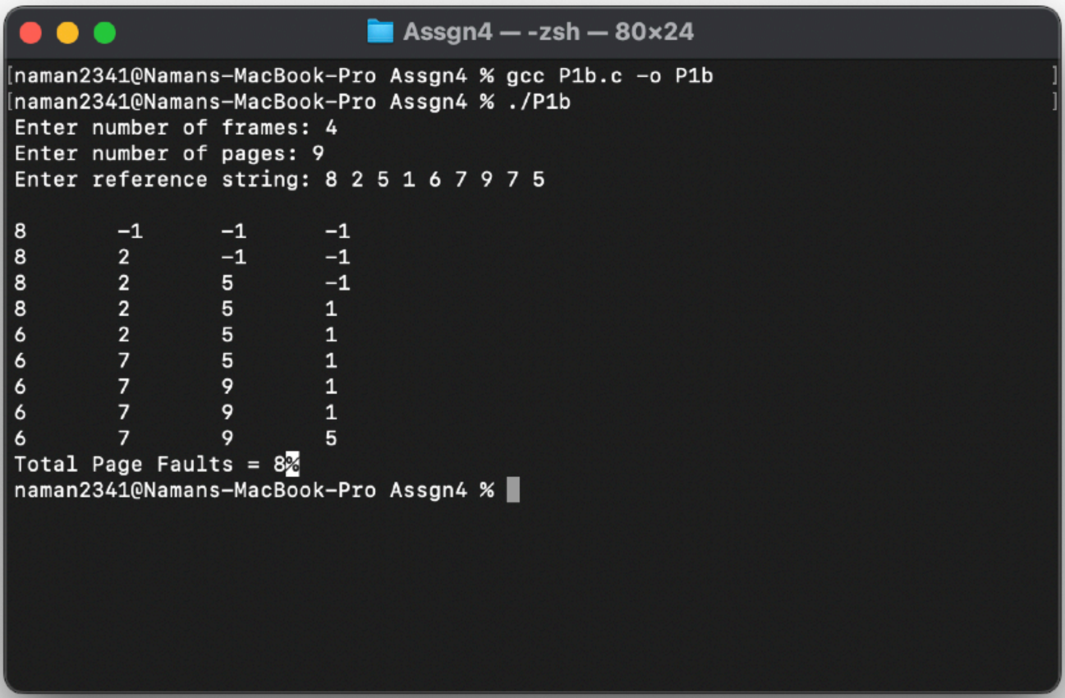
Source Code

```c
//C program for LRU replacement algorithm implementation
#include <stdio.h>
int findLRU(int time[], int n)
{
   int i, minimum = time[0], pos = 0;
   for (i = 1; i < n; ++i)
   {
      if (time[i] < minimum)
      {
         minimum = time[i];
pos = i; }
}
return pos; }

int main() {
   int no_of_frames, no_of_pages, frames[10], pages[30],
counter = 0, time[10], flag1,
flag2, i, j, pos, faults = 0;
   printf("Enter number of frames: ");
   scanf("%d", &no_of_frames);
   printf("Enter number of pages: ");
   scanf("%d", &no_of_pages);
   printf("Enter reference string: ");
   for (i = 0; i < no_of_pages; ++i)
   {
      scanf("%d", &pages[i]);
   }
   for (i = 0; i < no_of_frames; ++i)
   {
      frames[i] = -1;
   }
   for (i = 0; i < no_of_pages; ++i)
   {
      flag1 = flag2 = 0;
      for (j = 0; j < no_of_frames; ++j)
      {
         if (frames[j] == pages[i])
         {
            counter++;
            time[j] = counter;
            flag1 = flag2 = 1;
            break;
} }
      if (flag1 == 0)
      {
         for (j = 0; j < no_of_frames; ++j)
         {
            if (frames[j] == -1)
            {
               counter++;
               faults++;
               frames[j] = pages[i];
               time[j] = counter;
               flag2 = 1;
               break;
} }
}
      if (flag2 == 0)
      {
         pos = findLRU(time, no_of_frames);
         counter++;
         faults++;
         frames[pos] = pages[i];
         time[pos] = counter;
      }
      printf("\n");

for (j = 0; j < no_of_frames; ++j)
      {
         printf("%d\t", frames[j]);
      }
}
   printf("\nTotal Page Faults = %d", faults);
return 0; }
```

Output Screenshot



```
[naman2341@Namans-MacBook-Pro Assgn4 % gcc P1b.c -o P1b
[naman2341@Namans-MacBook-Pro Assgn4 % ./P1b
Enter number of frames: 4
Enter number of pages: 9
Enter reference string: 8 2 5 1 6 7 9 7 5

8       -1      -1      -1
8       2       -1      -1
8       2       5       -1
8       2       5       1
6       2       5       1
6       7       5       1
6       7       9       1
6       7       9       1
6       7       9       5
Total Page Faults = 8%
naman2341@Namans-MacBook-Pro Assgn4 %
```