LAB 10

QUESTION: Write a C program to simulate paging technique of memory management.

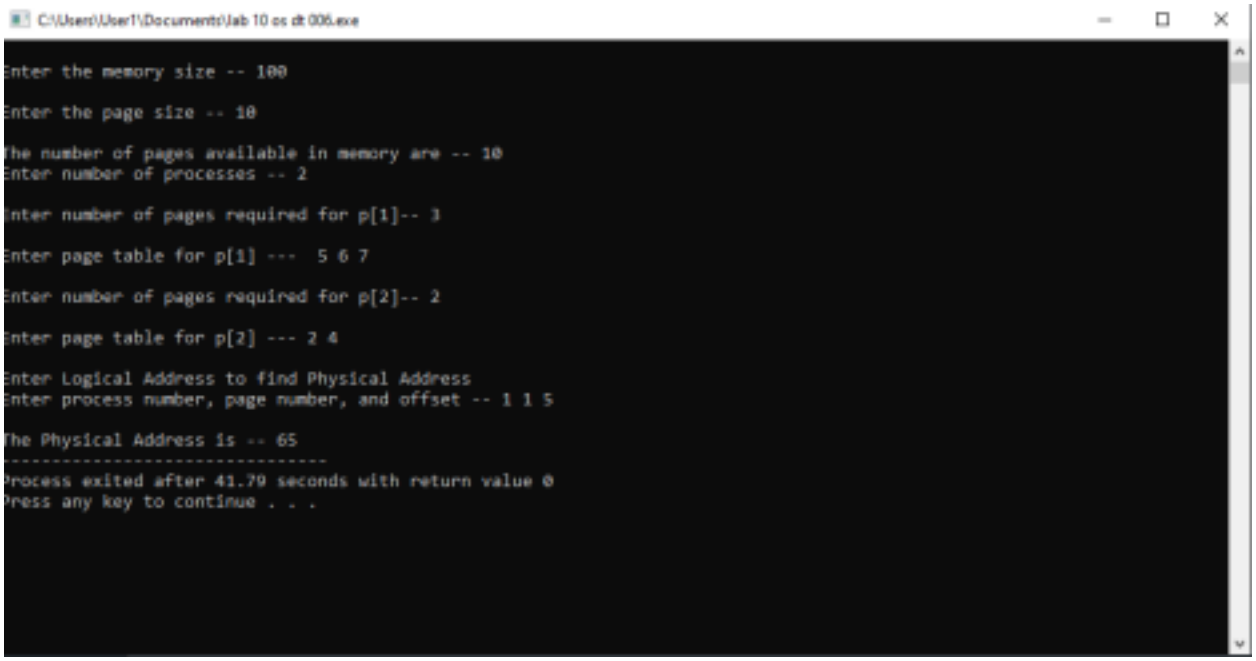
ANSWER:

CODE:

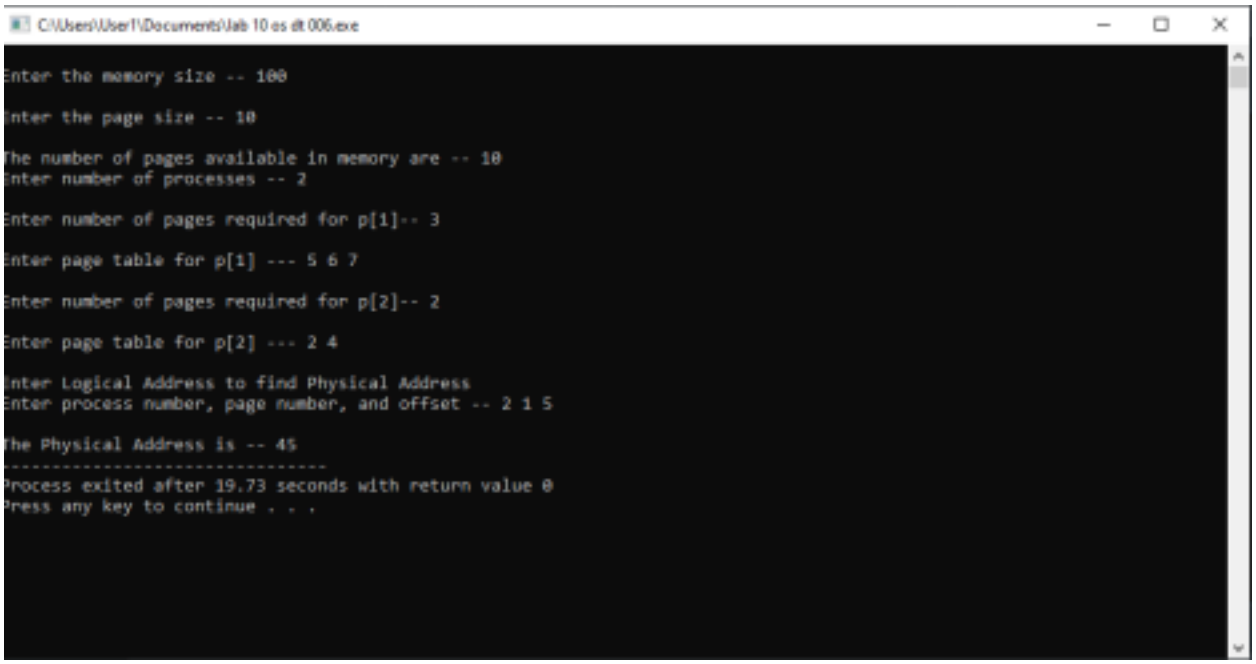
```
#include <stdio.h>
int main() {
 int ms, ps, nop, np, rempages, i, j, x, y, pa, offset;
 int s[10], fno[10][20];
 printf("\nEnter the memory size -- ");
 scanf("%d", &ms);
 printf("\nEnter the page size -- ");
 scanf("%d", &ps);
 nop = ms / ps;
 printf("\nThe number of pages available in memory are -- %d", nop);
printf("\nEnter number of processes -- ");
 scanf("%d", &np);
 rempages = nop;
 for(i = 1; i <= np; i++) {
 printf("\nEnter number of pages required for p[%d]-- ", i);
scanf("%d", &s[i]);
 if(s[i] > rempages) {
 printf("\nMemory is Full");
 break;
 }
 rempages = rempages - s[i];
 printf("\nEnter page table for p[%d] --- ", i);
```

```
 for(j = 0; j < s[i]; j++) {
 scanf("%d", &fno[i][j]);
 }
 }
 printf("\nEnter Logical Address to find Physical Address ");
printf("\nEnter process number, page number, and offset -- ");
scanf("%d %d %d", &x, &y, &offset);
 if(x > np || y >= s[x] || offset >= ps) {
 printf("\nInvalid Process or Page Number or Offset");  }
else {
 pa = fno[x][y] * ps + offset;
 printf("\nThe Physical Address is -- %d", pa);  }
 return 0;
}
```

OUTPUT:



Course: OPERATING SYSTEMS
Course code: CT-353



Name: Hiba Kazmi Roll no:DT-22025