```c
#include<stdio.h>

struct LRU
{
    int value;
    int index;
};

int valuePresentInCache(int V,struct LRU *LRUCache,int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        if(LRUCache[i].value==V)
            return i;

    }
    return -1;
}

int indexOfEmptyLoc(struct LRU *LRUCache, int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        if(LRUCache[i].value==-1)
            return i;
    }
    return -1;
}

int leastIndexValue(struct LRU *LRUCache,int n)
{
    int min=LRUCache[0].index;
    int minIndex=0;
    int i=1;
    while(i<n)
    {
        if(LRUCache[i].index<min)
        {
            min=LRUCache[i].index;
            minIndex=i;
        }
        i=i+1;

    }
    return minIndex;
}


void main()
{
    int PAGE[]={4,3,2,1,2,1,5,6,7,8,1,1,1,4,6,4,7,7,3,7,3,6,25,2};
    int size=sizeof(PAGE)/sizeof(int);

   int LRUSIZE=4;

    struct LRU LRUCache[LRUSIZE];

    // initialize the LRU
    int i;
    for(i=0;i<4;i++)
    {
        LRUCache[i].value=-1;
        LRUCache[i].index=-1;
    }
```

```
67
68      for(i=0;i<size;i++)
69      {
70          printf("\nIncoming Page is %d from index %d\n",PAGE[i],i);
71          int index=valuePresentInCache(PAGE[i],LRUCache,4);
72          //printf("present index %d\n",index);
73          if(index>=0) //page present in cache ,, update index value
74          {
75              LRUCache[index].index=i;
76          }
77          else
78          {
79              index=indexOfEmptyLoc(LRUCache,LRUSIZE);
80             // printf("empty loc index %d\n",index);
81              if(index>=0) // empty location present ,, insert page in cache and index
82              {
83                  LRUCache[index].value=PAGE[i];
84                  LRUCache[index].index=i;
85              }
86              else        // no empty loc, remove LRU page and insert CURRENT PAGE
87              {
88                          // find index of LRU page in LRUCache using Least value of LRU index;
89
90                  index=leastIndexValue(LRUCache,LRUSIZE);
91                 // printf("Minimim index %d\n",index);
92                  LRUCache[index].value=PAGE[i];
93                  LRUCache[index].index=i;
94              }
95          }
96
97
98  printf("\nCURRENT STATE OF LRU\n");
99  int m=0;
100  for(m=0;m<LRUSIZE;m++)
101  {
102      printf("PAGE NO: %d        INDEX: %d\n",LRUCache[m].value,LRUCache[m].index);
103  }
104
105
106     }
107  printf("\n\n Final State Of LRU\n");
108          for(i=0;i<LRUSIZE;i++)
109              printf("PAGE NO: %d        INDEX: %d\n",LRUCache[i].value,LRUCache[i].index);
110
111
112  }
```