Program Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<limits.h>

int *get_page_string(int n){
   int *ps = (int*)malloc(sizeof(int)*n);
   int prev = (int) ' ';
   int num;
   for(int i=0; i<n;i++){
      scanf("%d",ps+i);
   }
   return ps;
}

void show_page_frame(int* pf, int fr, int p){
   // p is the index in page string where the fault occured.
   char start[] = "|-----";
   char end[]   = "|-----|";
   int n = fr<p ? fr: p;

   printf("\n");
   for(int i=0; i<fr; i++){
      if(i!= fr-1)
      printf("%s", start);
      else
      printf("%s\n", end);
   }
   for(int i=0;i<fr;i++){
      if(i<=n){
         printf("|%5d",pf[i]);
      }
      else
         printf("|%5s"," ");
   }
   printf("|\n");
   for(int i=0; i<fr; i++){
      if(i!= fr-1)
      printf("%s", start);
      else
      printf("%s\n", end);
   }
}

int fifo(int *ps, int n, int fr){
   int first_in = 0, fault = 0, fl=0,j,l,i;

   int *page_frame = (int*) malloc(sizeof(int)*fr);

   for(j=0;j<fr;j++) page_frame[j] = -1;

   for(i = 0; i< n; i++){
      int fflag = 1; // assume page fault occures
      if(fr>fl){  // page frame is not filled
         for(j=0;j<fl;j++){
            if(page_frame[j]==ps[i]){
```

```
                fflag = 0;
                break;
            }
        }
        if(fflag){
            page_frame[fl] = ps[i];
            fault++;
            show_page_frame(page_frame,fr,fl);
            fl++;
        }
    }
    else{
        for(j=0;j<fr;j++){
            if(page_frame[j]==ps[i]){
                fflag=0;
                break;
            }
        }
        if (fflag){
            //int pos = get_lru_index(ps,n, page_frame, fr, i);
            page_frame[first_in] = ps[i];
            first_in = (first_in+1)%fr;
            fault++;
            show_page_frame(page_frame,fr,i);
        }
    }
}

free(page_frame);
return fault;
}

int get_lru_index(int* ps, int n,int* pf, int fr, int i){
    int j,k=0,l=0;
    int pos=0;
    int matched[fr];
    for(j=0; j<fr; j++) matched[j] = 0;
    for(j=i-1; j >= 0 && k < fr; j--){       //iterate until all the elements in the frame string are found
        for(l=0; l<fr; l++){
            if(ps[j] == pf[l] && !matched[l]){ // check if the page is in the page frame
                //printf("Matched %d at %d\n",pf[l],j);
                pos = l;      //Position of the page to be replaced.
                k++;
                matched[l] = 1;
                break;
            }
        }
    }
    //printf("\nPostion = %d\n",pos);
    return pos;
}

int lru(int *ps, int n, int fr){
    int i,j,fault = 0,fl=0;
    int *page_frame = (int*) malloc (sizeof(int)*fr);
    for(j=0;j<fr;j++) page_frame[j] = -1;
```

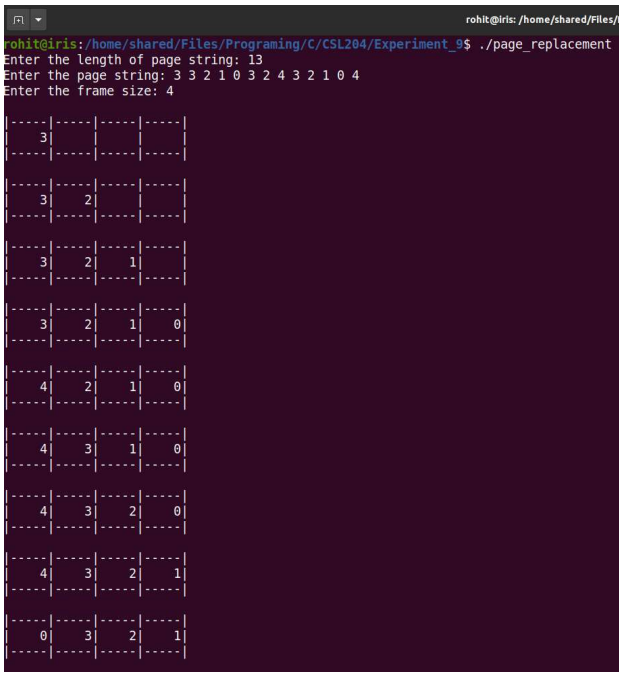
```
    for(i = 0; i< n; i++){
        int fflag = 1; // assume page fault occures
        if(fr>fl){  // page frame is not filled
            for(j=0;j<i;j++){
                if(page_frame[j]==ps[i]){
                    fflag = 0;
                    break;
                }
            }
            if(fflag){
                page_frame[fl] = ps[i];
                fault++;
                show_page_frame(page_frame,fr,fl);
                fl++;
            }
        }
        else{
            for(j=0;j<fr;j++){
                if(page_frame[j]==ps[i]){
                    fflag=0;
                    break;
                }
            }
            if (fflag){
                int pos = get_lru_index(ps,n, page_frame, fr, i);
                page_frame[pos] = ps[i];
                fault++;
                show_page_frame(page_frame,fr,i);
            }
        }
    }

    free(page_frame);
    return fault;
}

int get_lfu_index(int* ps, int n, int* pf, int fr, int i, int* freq_array){
    int pos = -1;
    int min = INT_MAX;

    for(int j=0; j<fr; j++){
        if( freq_array[j] < min){
            min = freq_array[j];
            //printf("Min freq: %d of %d\n",pf[j],min);
            pos = j;
        }
    }

    return pos;
}

int lfu(int* ps, int n, int fr){
    int i,j,k,fault = 0,fl = 0;
    int *page_frame = (int*)malloc(sizeof(int)*fr);
    int *freq_arr = (int*)malloc (sizeof(int)*fr); for(int i = 0;i<fr;i++) freq_arr[i]=0;

    for(i=0;i<fr;i++) page_frame[i] = -1;
```

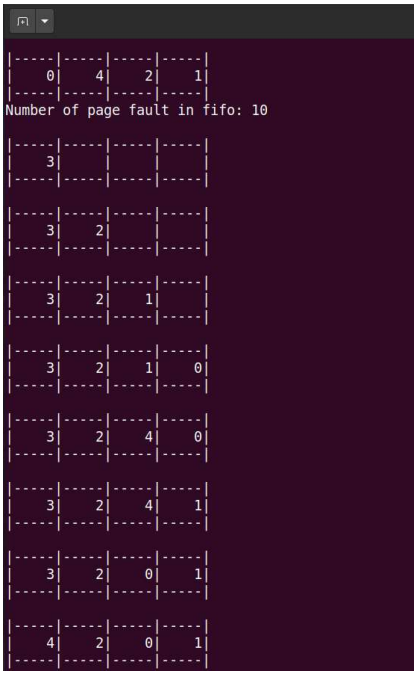
```
    for(i=0;i<n;i++){ // Iterate through the page string
        int fflag = 1 ;  // Assume that a fault had occured.
        if(fr>fl){  // There is still some free space in the page frame
            for(j=0;j<fr;j++){
                if(ps[i]==page_frame[j]){
                    fflag = 0;  // The page in question is in the page frame.
                    freq_arr[j] = freq_arr[j]+1;
                    break;
                }
            }
            if(fflag){  // If the page is not found.
                page_frame[fl] = ps[i];
                freq_arr[fl] = 1;
                fault++;
                show_page_frame(page_frame,fr,fl);
                fl++;
            }
        }
        else{ // Page frame is full
            for(j=0;j<fr;j++){
                if(ps[i]==page_frame[j]){
                    fflag = 0;  // The page in question is in the page frame.
                    freq_arr[j] = freq_arr[j]+1;
                    break;
                }
            }
            if(fflag){
                int pos = get_lfu_index(ps,n,page_frame, fr, i,freq_arr);
                for(int p=pos; p<fr; p++){
                    page_frame[p] = page_frame[p+1];
                    freq_arr[p] = freq_arr[p+1];
                }
                page_frame[fr-1] = ps[i];
                freq_arr[fr-1] = 1;
                fault++;
                show_page_frame(page_frame,fr,i);
            }
        }
    }

    free(freq_arr);
    free(page_frame);
    return fault;

}

int main(){
    int n,frames;
    printf("Enter the length of page string: ");
    scanf("%d",&n);
    printf("Enter the page string: ");

    int *page_string = get_page_string(n);
    printf("Enter the frame size: ");
    scanf("%d",&frames);

    int fifo_f = fifo(page_string,n,frames);
```

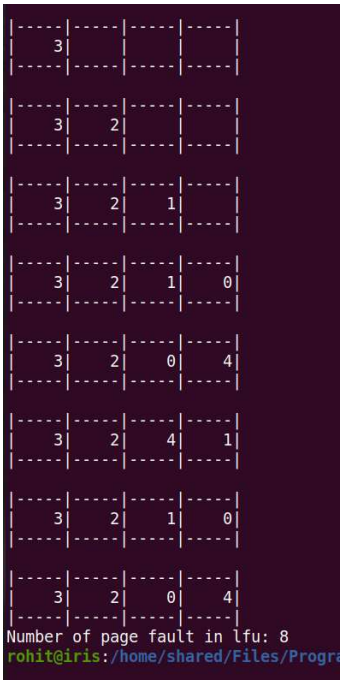
```
    printf("Number of page fault in fifo: %d\n",fifo_f);

    int lru_f = lru(page_string,n,frames);
    printf("Number of page fault in lru: %d\n",lru_f);

    int lfu_f = lfu(page_string,n,frames);
    printf("Number of page fault in lfu: %d\n",lfu_f);

    free (page_string);
    return 0;
}
```

Screenshots

Input 1:







Input 2: