To stimulate page Replacement Algorithm

1) FIFO

```
# include <std.o.h>
# define frame size 3
void fifo (int reference_string [], int length ) {
 int frames [frame_size ];
 int front = 0;
 for (i = 0; i < frame_size ; i++)
    frame [i] = -1 ; }

int page faults = 0;
  for (i = 0; i< length ; i++)
    int page = reference_string[i];
    int found = 0;
 for (j = 0; j < frame_size ; j++)
   if (frame [j] == page){
     found = 1;
     break ;
 }}
 if (! found ){
   frames [front ] = page;
   front = (front +1) % frame size ;
    page faults ++ ; }
 printf ("page %d : [" page);
   for (j = 0; j < frame size ; j++)
     if (frames [j] == -1)
          printf (" - ");
 } else {
   printf (" %d", frames [j]);
 }}
   printf ("]\n"); }
   printf (" Total page faults %d \n" page faults)
}
```

```
int main (){
  int length;
  printf ("Enter the length of reference string:");
  scanf ("%d", &length);
  int reference_string [length];
  printf ("Enter the reference string \n");
   for (i=0; i<leng ; i++){
     scanf ("%d", &reference_string [i]);
   }
  fifo (reference_string, length);
   return 0;
}.
```

2) Output:

```
Enter the no of pages : 12
Enter the reference string: 2 3 2 1 5 2 4 5 3 2 5 2

 2  -1  -1
 2   3  -1
 2   3  -1
 2   3   1
 5   3   1
 5   2   1
 5   2   4
 5   2   4
 3   2   4
 3   2   4
 3   5   4
 3   5   2

No of page faults : 9
```

2) LRU:

```
#include <stdi
#define frame
Void free ( int
  int frames
  int free
  for (i=0;
    frames [i] =
  int page_f
  for (i=0 :
    frames [i]
  int page_
  for (j=0;
    if (fram
    found =
    free_ind
    break;
  printf (" pa
    if (fram
      printf
    }
  else {
    printf
  } }
  printf (" ]
  printf (" To
}
int main (
  int len
  printf
  scanf (
```

2) LRU:

```
#include <stdio.h>
#define frame-size 3
Void free (int reference-string [], int length) {
  int frames [frame-size];
  int tree-index;
 for (i=0; i< frames-size ; i++)
   frames [i]=-1; }
 int page-faults =0;
 for (i=0 ; i< frames-size ; i++) {
   frames [i] =-1; }
  int page-faults =0;
 for (j=0; j<frame-size ; j++) {
   if (frames [j] == page) {
   found = 1
   tree-index = j
   break ; }}
printf ("page %d : [", page);
  if (frames [j] == -1) {
    printf ("-");
    }
 else {
  printf ("%d", frames[j]);
 }}
printf ("]\n"); }
printf ("Total page faults : %d \n", pagefaults)

int main () {
  int length;
  printf ("Enter the leng of referencestring :");
  Scanf ("%d", & length);
```

```
int reference_string[length];
printf("Enter the reference string:\n");
for (i=0; i<length; i++){
  scanf("%d", &reference_string[i]);
}
printf("LRU Page Replacement\n");
return 0;
}
```

Result:

```
Enter the no of pages: 12
Enter the reference string: 2 3 2 1 5 2 4 5 3 2 5 2

2  -1  -1
2   3  -1
2   3  -1
2   3   1
2   5   1
2   5   1
2   5   4
2   5   4
3   5   4
3   5   2
3   5   2
3   5   2
```

No of page faults: 7

3) Optimal:

```
#include <stdio.h>
#include <limits.h>
#define frame_size 3
int find_optimal (int reference_string[], int frames[]);
  int start , int len ;
   int index = -1 ;
    int farthest = start ;
    for (i=0 ; i< frame_size ; i++){
     for (j=start ; j< length ; j++) {
       if (frames [i] == reference_string [j]) ;
          if (j > farthest) {
              farthest = j ;
              index = i ; }
          break ; }}
Void optimal (int reference_string [], int length).
  int frames [frame_size] ;
 for (i = 0 ; i < frame_size ; i++) {
   frames [i] = -1 ;
 int page_faults = 0 ;
  for (i=0 ; i< len ; i++){
     int found=0 ;
  for (j=0 ; j < frame_size ; j++){
    if (frames[j] == page){
       found = 1 ;
       break ;
  }}
printf ("Total Page faults: %d\n", page-faults);
int main() {
   int length;
   printf ("Enter leng reference string");
   Scanf ("%d", & len);
   printf ("Enter reference string \n");
      Scanf ("%d", & reference-string [i])
   }
   printf ("Optimal page replacement \n");
   optimal (reference-string, leng);
   return 0;
}
```

Output:

```
Enter the length: 12
Enter the reference string: 1 2 3 4 1 2 5 1 2 3 4 5
     1  -1  -1
     1   2  -1
     1   2   3
     1   2   4
     1   2   4
     1   2   4
     1   2   5
     1   2   5
     1   2   5
     3   2   5
     4   2   5
     4   2   5
     4   2   5

No. of page faults : 7
 page fault rate = 58.33
```

10/10

28/8/23

To stimulate d

1) fcfs

```
# include <std
# include <stdl
Void fcfs (int r
   int total- m
for (i=1;  i<n;
  total movement
   printf (" Tot
}
int main() {
   int n, head
   printf (" Ent
   Scanf ("%d
   int request
   printf (" En
    for (i=0;
      Scanf ("
   }
   printf ("Ente
   Scanf ("%d
     fcfs (req
    return 0;
  }
```

output

```
Enter no of
Enter request
Enter initial
Total head
```