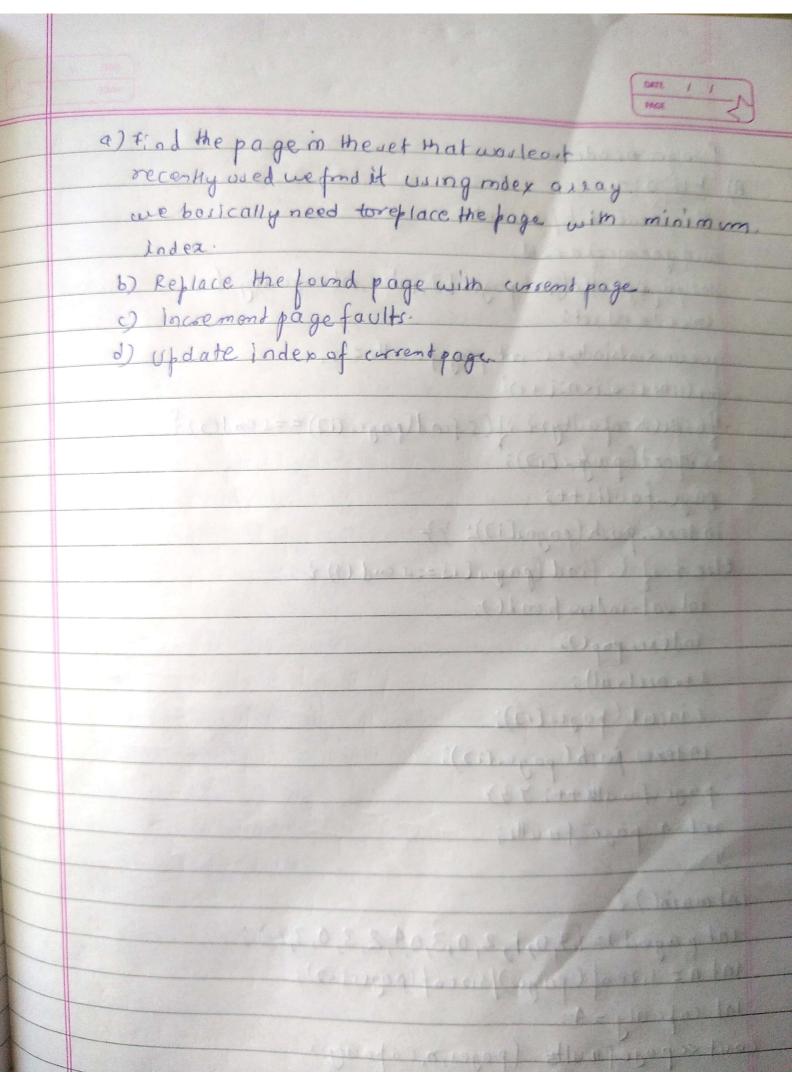
	Experiment-87 Peepesh Toux.
	Page Replacement Algorithm 18075017
	Arm (1) PAGE
	Aim:-
	write a C program to stimulate the following
	page replacement algorithm
	a) FIFO
	b) Optimal Page Replacement
	c) LRU.
	supplies and which is a first form of the file
	Theory = - 1
a)	FIFO: - Al sold services and services are services are services and services are services and services are services are services and services are services and services are services are services and services are services are services and se
	This is the simplest pages eplacement technique in this Os
k	eep trocks of all pages in the memory in a quede older
þ	age is in the front of queve, when a page is needed to be
	ellaced page in front of greve is relected for removal
	A CLOS OF THE CONTRACTOR OF THE PARTY OF THE
	algrithm of fifo:
	1) - Start haveing the pages
	i) if set hold less pages than capacity
	a) insect page mb set one by one (nh) size of set
	reaches capacity de all pagerequest are proceed.
	b) simultaneously maintain au pages in the que ve to
	perform fifo
	() incoment page fault
	1) ele marie la sala de la sala d
	if current page is present in set, do nothing
	Else
	o) Remove the first page from the queue to acit was
	first to be entered in the memory the correct
	5) Replace the first page in the queue with current
	page m the string
	Store current page in guerre
	e) return page fault
	Scanned with CamScanner

	MG A
(b)	Of-Kmal Page replacement:-
	in OS, whenever this algorithm OS will replace the page may
	will not be used for the longest period of time in future
	algorithm:
	la every reference we do following:
	i) rejered page is along ody present, in crement hit count
	2) if not present find if page that is never refesence in
	future if such a page exsist replace this page with new
	rage if no such page exsut and the page that is reference for thest in future helplace this page with new
	pager
	male and holder of the property of the latter of the latte
c)	LRU:-
	Least Recenty used (LRU) Algorithm is a
	greedy algorithm wonceded where the page
	to be replaced is least recontly used The idea is based
	on locally of reference.
	algorithm:
	let capacity to the number of pages that memory can
	hold let set be the current set of pages in memory
	1) start traversing the pages
	Dif set holds less jages than capacity
	O) insert pages into the set, do nothing one by one
	Etre until the size of set reacher capacity or all
	6) Prodpag e reg vert ave fulfilled.
	b) stmultaneously maintain the event occurred
	lader of each page in a map called indexe
3	in Else
-	
-	if current page is present in set, donothing
	Elje The page of t
	Scanned with CamScanne



```
Programcode:-
#include (bits/stdc+p.h7
 wing nomespaces to:
 Intragefaults (intrages [], into, int capacity) [
 mordered set;
 que ve l'intrinderesi int page faulti = 0;
 for (Intizoiknii+e)
 1/6.518e(Xcapacity) & if (s.find(pages (i)) == s-end()) {
  s.insext(pages[i));
  page-faults++;
  Indexesposh(pages[1)): 47
 else { if (sfird (pages [i] = = s. end ()) {
   int val=indexs front();
    Indexerpop()
    s. exose(val);
    s. insert (pages [i]);
  Indexes fush (pages(is);
   page foult to; y3)
   rehm page fault;
 intmain() {
 int pages[]= {7,0,1,2,0,3,0,4,2,3,0,3,23;
 Int n = size of (pager) size of (pager (0))
 int copacity = 4:
cout exposefaults (poses, n, Capacity);
output-7
```

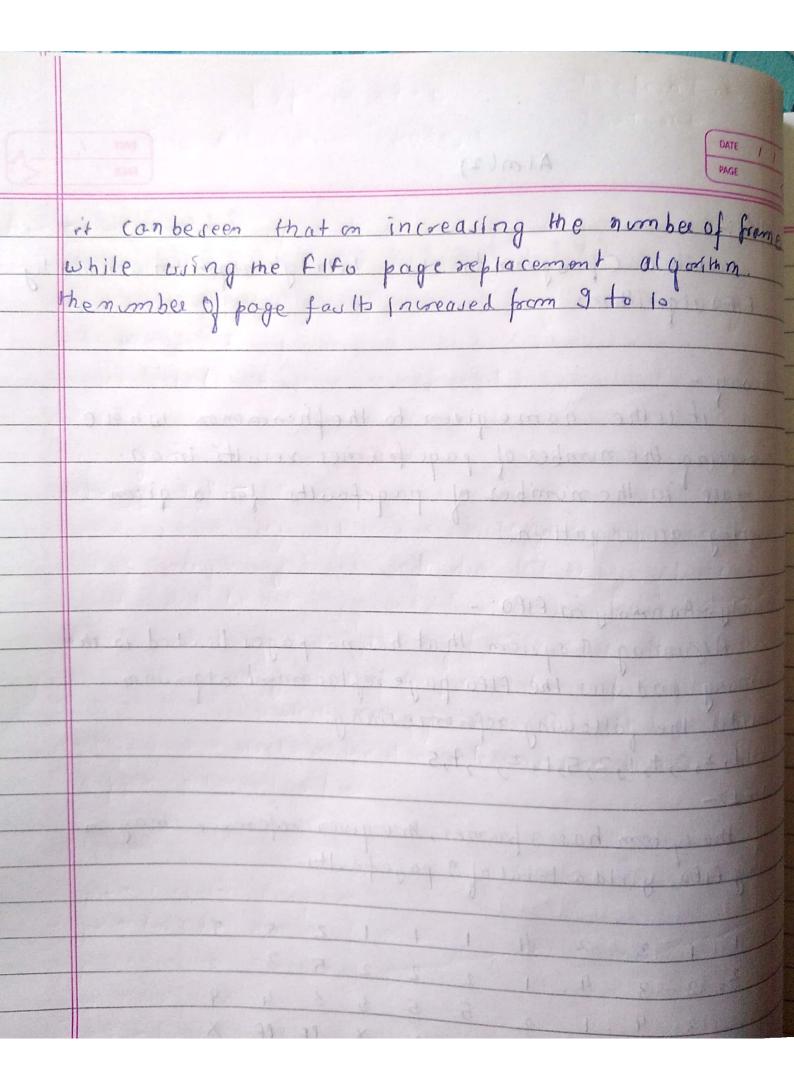
```
b) of timal page replacement algorithm
  #includes blts stdc++. hz.
                                   8 0 2 1 6
   using nomespacestd;
  using nomespacestd:
   boolsearch (int key, rectorkints & fr) {
   for (inti=0) icfo-size(); i++) if (focio) == key) return true;
  return falses y.
  int predict (Int pgcz, rector (int) fr int pn, int index) {
  intres=-1, farthest=index;
  for (int i=0; ixfo.size(); i++) { int;
 tor (j=index)jkpn:j+P) }
   (FELI] == POCIJ) { if (j > for mest) {
  famest=j:res=j; y break; } }
  (j==pn) returnd; y
  retion (res == -1)? O: res :
 roid of time page (int pg [], inten, int (n) ?
   rector sinto /s
  int hit = 0;
  for (int i=0; i<pn; i++) {
  if ( search (pg[i], fo)) {
      hit++;
     (mhnue;
  (fo. size() <fn) fr. push_bock(pg[I]);
  else 1
    int j = predict (pg, fr, pn, i+1);
   Je (j) = 19 [i) 343
 gout se" No of hits = " < hit Kendli
 Gost K " No of misses = " K pn-hot K end);
```

	The second design of the second second
int pg [] = 17, 0, 1, 2, 0	4. 2.3.0.3,23;
int pg [] = 17, 0, 1, 2, 0 int pg [] = 17, 0, 1, 2, 0 int pn = size of (pa) / size o	30919
int pg ((og)/sizeo	F(Pg[0])
lint by side	
10th	May Stanford
aptimal page (88, pn, 171)	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
retuno: 3	
	7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
output:- Malanta a Ha	A SCIOIX MALE DE LE CARLO DE LA CONTRACTOR DEL CONTRACTOR DE LA CONTRACTOR DE LA CONTRACTOR DE LA CONTRACTOR
Poothit=7	
Prof misser=6	· 140 ((++1.0) 9-12-12 12 103
700 11536	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	(+2011 - 1 1 36 F ES 69 82)
	1 1 for som 1 1 1 2 19 00 12
	Y I LIVE TO COM
	18:00 11-2218
3 (0)40	medical films for Tologo
	27510
	1- (+++ 2-4-7) 103
	LIGHT OF THE PARTY
	Scanned with CamScanned

```
C) LPU:-
  # include < bits | stdc++h7.
   using nomes pacestd:
  int page foults (int pages[]intr, int copocity)
    mordered setkint 75;
   mardered map (mt, intrindexes;
   in - page-faults = 0;
  for (int 1=0: in)[++]S
   if (s. sizel) (capacity) S
  il (s.fmd (pages [is) == s.endl))
   { s.insert (pogestis);
      page faulti++;3
  indexes[pagesli]=isy
   il (s-find (pages (i)) == s-end ())
    int lou= INT_MAX, val;
  tor (auto it = 8 begin Osit 1 = 8. end (): it++ )
  il (indexes [*it] < lav).
    Iru = indexes [*it]
      ral = *it; 33
  S. er ase (val):
  S. Invert (pages (i));
  page faults tt;
  Indexes[pages[i]]=i33
  rehan page faults; y
```

imporint main() {	
Intrages[]= (7,0,1,2,0,3,0,4,2,3,0,3,2):	
Int n = Size of (pages) / size of (pages (0));	
Intapacity=4:	4
Cout (cpagefaults (pages, n, capacity);	No.
rebrao:	100
B	011
baffat - 6	SEI
2 Copporate 3	
(Chartes & Chispage	) 60
(Pageoffin):	
£:++3101	63
2 - 1 = 1 0 0 0 0	org
LE Proposition de la company	
TO THE XAPETYT	- 0
- The the state of	133
(00/2/1)	
L)j*Jiaxib	OF
	1100
	Cla
	200
	-12
	1
Scanned with CamSca	anner

	Alm(2) PAGE	1
-	Almino shows the galleroot as the state and show	
	write a Cprogram to show belady's afferithm an	mal
	in FIFO algorithm:	and 19
	Theory:	
	it is the name given to the phenomenon when	6
	increasing the number of page frames results in an.	
	increase in the number of pagefoults for a given	
	memory access jattern.	
	Belady's Anomoly in FIFO: -	
	Assuming a system that has no pages loaded in	the
	memory and uses the fife page replacement algorithm	
1	Consider the following reference string:	
-	1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5	
	(avel)-	
	if the system has 3 frames, the given reference string	CM .
-	bring fifo. yield a total of 9 pagetaults.	
	111234111255	
	2 2 3 4 1 2 2 2 5 3 3	
	3 4 1 2 5 5 5 3 4 4	
	be be be be be be be x x be be x	
	Case 2:-	
	If the system has 4 frames. The given reference string	
	on using fifs page replacement algorithm yield a hotal	ofia
	pagefasts	
	1111234512	
-	2 2 2 2 2 3 4 5 1 2 3	
	3 3 3 3 4 5 1 2 3 4	
-	4 4 4 5 1 2 3 4 5	



```
Plogramiode
 # includes stdio.ho
 ot main(){
 roid page faults (mt frame size, int ref, int len)
  { int arr = new int [frome size];
 for (int i=0; ix frome_size; i++)
 011[1]=-1;7
 Int cnt=0; Int stort=0: Intflog; intelm;
 for (inti=o;ixlen; i++) { if (elm==arr[j]) { | lag=1; break; y 3
   | (start < frame size) {
 1 (110g==0) [
 arr [start] = elmi
stort ++; f
else il (rtort == frame size) {
 arr[0]=elm; stort=1;4 (n+++;33
printf (" when nim ber of fromes ore : "Id " frame size);
 pointf ("the number of page faults is tod In", cnt);
 nt main ()
 2 int ref[] = {1,2,3,4,1,2,5,1,2,3,4,53;
 intlen=sizeof (ref) size of (ref(a));
 interframe-size=3;
 ragefaulte frame size, ref, len);
 frome size ++;
 pagefoults (fromesize, ret, len);
output:-
when the number of fromes are 3, the number of page faults is: 9
when the number of former are 4, the number of page fault is? 1=
```