**Q26)**

These sites contain algorithms as well as codes in C

**best fit-**

[**https://prepinsta.com/operating-systems/page-replacement-algorithms/best-fit/**](https://prepinsta.com/operating-systems/page-replacement-algorithms/best-fit/)

**First fit-**

[**https://prepinsta.com/operating-systems/page-replacement-algorithms/first-fit/**](https://prepinsta.com/operating-systems/page-replacement-algorithms/first-fit/)

**Worst fit-**

[**https://prepinsta.com/operating-systems/page-replacement-algorithms/worst-fit/**](https://prepinsta.com/operating-systems/page-replacement-algorithms/worst-fit/)

**Q27)**

**FIFO**

Algorithm for FIFO Page Replacement

* Step 1. Start to traverse the pages.
* Step 2. If the memory holds fewer pages, then the capacity else goes to step 5.
* Step 3. Push pages in the queue one at a time until the queue reaches its maximum capacity or all page requests are fulfilled.
* Step 4. If the current page is present in the memory, do nothing.
* Step 5. Else, pop the topmost page from the queue as it was inserted first.
* Step 6. Replace the topmost page with the current page from the string.
* Step 7. Increment the page faults.
* Step 8. Stop
* #include <stdio.h>
* int main()
* {
* int referenceString[10], pageFaults = 0, m, n, s, pages, frames;
* printf("\nEnter the number of Pages:\t");
* scanf("%d", &pages);
* printf("\nEnter reference string values:\n");
* int(m = 0; m < pages; m++)
* {
* printf("Value No. [%d]:\t", m + 1);
* scanf("%d", &referenceString[m]);
* }
* printf("\n What are the total number of frames:\t");
* {
* scanf("%d", &frames);
* }
* inttemp[frames];
* for(m = 0; m < frames; m++)
* {
* temp[m] = -1;
* }
* for(m = 0; m < pages; m++)
* {
* s = 0;
* for(n = 0; n < frames; n++)
* {
* if(referenceString[m] == temp[n])
* {
* s++;
* pageFaults--;
* }
* }
* pageFaults++;
* if((pageFaults <= frames) && (s == 0))
* {
* temp[m] = referenceString[m];
* }
* else if(s == 0)
* {
* temp[(pageFaults - 1) % frames] = referenceString[m];
* }
* printf("\n");
* for(n = 0; n < frames; n++)
* {
* printf("%d\t", temp[n]);
* }
* }
* printf("\nTotal Page Faults:\t%d\n", pageFaults);
* return 0;
* }

Enter the number of Pages: 5

Enter reference string values:

Value No. [1]: 4

Value No. [2]: 1

Value No. [3]: 2

Value No. [4]: 4

Value No. [5]: 5

What are the total number of frames: 3

4 -1 -1

4 1 -1

4 1 2

4 1 2

5 1 2

Total number of page faults: 4

**LRU**

Step 1. Start the process

Step 2. Declare the page size

Step 3. Determine the number of pages to be inserted.

Step 4. Get the value.

Step 5. Declare the counter and stack value.

Step 6. Choose the least recently used page by the counter value.

Step 7. Stack them as per the selection.

Step 8. Display the values.

Step 9. Terminate the process.

#include

main()

{

int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];

printf("Enter no of pages for which you want to claculate page faults:");

scanf("%d",&n);

printf("Enter reference string:");

for(i=0;i<n;i++)

scanf("%d",&p[i]);

printf("Enter no of frames:");

scanf("%d",&f);

q[k]=p[k];

printf("\n\t%d\n",q[k]);

c++;

k++;

for(i=1;i<n;i++)

            {

                        c1=0;

                        for(j=0;j<f;j++)

                        {

                                    if(p[i]!=q[j])

                                    c1++;

                        }

                        if(c1==f)

                        {

                                    c++;

                                    if(k<f)

                                    {

                                                q[k]=p[i];

                                                k++;

                                                for(j=0;j<k;j++)

                                                printf("\t%d",q[j]);

                                                printf("\n");

                                    }

                                    else

                                    {

                                                for(r=0;r<f;r++)

                                                {

                                                            c2[r]=0;

                                                            for(j=i-1;j<n;j--)

                                                            {

                                                            if(q[r]!=p[j])

                                                            c2[r]++;

                                                            else

                                                            break;

                                                }

                                    }

                                    for(r=0;r<f;r++)

                                     b[r]=c2[r];

                                    for(r=0;r<f;r++)

                                    {

                                                for(j=r;j<f;j++)

                                                {

                                                            if(b[r]<b[j])

                                                            {

                                                                        t=b[r];

                                                                        b[r]=b[j];

                                                                        b[j]=t;

                                                            }

                                                }

                                    }

                                    for(r=0;r<f;r++)

                                    {

                                                if(c2[r]==b[0])

                                                q[r]=p[i];

                                                printf("\t%d",q[r]);

                                    }

                                    printf("\n");

                        }

            }

}

printf("\nThe total number of page faults are %d",c);

}

Enter no of pages for which you want to calculate page faults: 10

Enter the reference string: 1 2 2 3 4 5 6 7 8 9

Enter number of frames: 3

1

1          2

1          2          2

3          4          5

6          7          8

9          7          8

9          7          8

9          6          7

9          6          2

1          6          2

Number of page faults is 10

**OPTIMAL**

**Step 1:** Push the first page in the stack as per the memory demand.

**Step 2:**  
Push the second page as per the memory demand.

**Step 3:**  
Push the third page until the memory is full.

**Step 4:**  
As the queue is full, the page which is least recently used is popped.

**Step 5:**  
repeat step 4 until the page demand continues and until the processing is over.

**Step 6:**  
Terminate the program.

#include

int main()

{

//variable declaration and initialization

int frames\_number, pages\_number, frames[10], pages[30], temp[10], flag1, flag2, flag3, i, j, k, pos, max, miss = 0;

//code to input the frame number

printf("Enter number of frames: ");

scanf("%d", & frames\_number);

//code to input number of pages

printf("Enter number of pages: ");

scanf("%d", &pages\_number);

//code to define reference string, page numbers, and frame numbers

printf("Enter page reference string: ");

for(i = 0; i < pages\_number; ++i){

scanf("%d", &pages[i]);

}

for(i = 0; i < frames\_number; ++i){

frames[i] = -1;

}

for(i = 0; i < pages\_number; ++i){

flag1 = flag2 = 0;

for(j = 0; j < frames\_number; ++j){

if(frames[j] == pages[i]){

flag1 = flag2 = 1;

break;

}

}

//definition of the flag at the starting of the string

if(flag1 == 0){

for(j = 0; j < frames\_number; ++j){

if(frames[j] == -1){

faults++;

frames[j] = pages[i];

flag2 = 1;

break;

}

}

}

// definition of the flag at the mid position

if(flag2 == 0){

flag3 =0;

for(j = 0; j < frames\_number; ++j){

temp[j] = -1;

for(k = i + 1; k < pages\_number; ++k){

if(frames[j] == pages[k]){

temp[j] = k;

break;

}

}

}

for(j = 0; j < frames\_number; ++j){

if(temp[j] == -1){

pos = j;

flag3 = 1;

break;

}

}

//definition of flag at the rear position

if(flag3 ==0){

max = temp[0];

pos = 0;

for(j = 1; j < frames\_number; ++j){ if(temp[j] > max){

max = temp[j];

pos = j;

}

}

}

frames[pos] = pages[i];

miss++;

}

printf("\n");

for(j = 0; j < frames\_number; ++j){

printf("%d\t", frames[j]);

}

}

printf("\n\nTotal Page miss = %d", miss);

return 0;

}

Enter the number of frames: 3

Enter the number of pages: 8

Enter page reference string: 4 5 7 4 3 5 7 5

4 -1 -1

4 5 -1

4 5 7

4 5 7

3 5 7

3 5 7

3 5 7

3 5 7