#include <iostream>

#include <new>

#include <queue>

using namespace std;

queue<char> fila;

int page\_faults = 0;

int string\_position = 0;

#define NUMBER\_OF\_PAGES 4

void init\_physical\_memory(char \*physical\_memory, char reference\_string[]);

bool has\_pages\_faults(char \*physical\_memory, char c);

void refresh\_memory(char \*physical\_memory, char new\_char);

void print\_physical\_memory(char \*physical\_memory);

int main()

{

char reference\_string[] = {'2', '3', '1', 'a', 'b', '3', '1', '1', 'c', 'd', '1', 'a'};

char \*physical\_memory = new (nothrow) char[NUMBER\_OF\_PAGES];

init\_physical\_memory(physical\_memory, reference\_string);

cout << "Memoria fisica inicial ";

print\_physical\_memory(physical\_memory);

cout << endl;

for (int i = string\_position; i < sizeof(reference\_string)/sizeof(\*reference\_string); i++)

{

if (has\_pages\_faults(physical\_memory, reference\_string[i]))

{

page\_faults++;

refresh\_memory(physical\_memory, reference\_string[i]);

print\_physical\_memory(physical\_memory);

cout << endl;

}

}

cout << "Numero de Pages Faults: " << page\_faults << endl;

}

void init\_physical\_memory(char \*physical\_memory, char reference\_string[])

{

for (int i = 0; i < NUMBER\_OF\_PAGES; i++)

{

physical\_memory[i] = reference\_string[i];

fila.push(reference\_string[i]);

page\_faults++;

string\_position++;

}

}

bool has\_pages\_faults(char \*physical\_memory, char c)

{

bool has = true;

for (int i = 0; i < NUMBER\_OF\_PAGES; i++)

{

if (physical\_memory[i] == c)

{

has = false;

}

}

return has;

}

void refresh\_memory(char \*physical\_memory, char new\_char)

{

for (int i = 0; i < NUMBER\_OF\_PAGES; i++)

{

if (physical\_memory[i] == fila.front())

{

physical\_memory[i] = new\_char;

fila.pop();

fila.push(new\_char);

return;

}

}

}

void print\_physical\_memory(char \*physical\_memory)

{

for (int i = 0; i < NUMBER\_OF\_PAGES; i++)

{

cout << " " << physical\_memory[i] << "";

}

}