

```

1  #include <iostream>
2  #include <new>
3  #include <queue>
4
5  using namespace std;
6
7  queue<char> fila;
8  int page_faults = 0;
9  int string_position = 0;
10 #define NUMBER_OF_PAGES 4
11
12 void init_physical_memory(char *physical_memory, char reference_string[]);
13 bool has_pages_faults(char *physical_memory, char c);
14 void refresh_memory(char *physical_memory, char new_char);
15 void print_physical_memory(char *physical_memory);
16
17 int main()
18 {
19     char reference_string[] = {'2', '3', '1', 'a', 'b', '3', '1', '1', 'c', 'd', '1', 'a'};
20     char *physical_memory = new (nothrow) char[NUMBER_OF_PAGES];
21
22     init_physical_memory(physical_memory, reference_string);
23
24     cout << "Memoria fisica inicial ";
25     print_physical_memory(physical_memory);
26     cout << endl;
27
28     for (int i = string_position; i < sizeof(reference_string)/sizeof(*
reference_string); i++)
29     {
30         if (has_pages_faults(physical_memory, reference_string[i]))
31         {
32             page_faults++;
33             refresh_memory(physical_memory, reference_string[i]);
34             print_physical_memory(physical_memory);
35             cout << endl;
36         }
37     }
38     cout << "Numero de Pages Faults: " << page_faults << endl;
39 }
40
41 void init_physical_memory(char *physical_memory, char reference_string[])
42 {
43     for (int i = 0; i < NUMBER_OF_PAGES; i++)
44     {
45         physical_memory[i] = reference_string[i];
46         fila.push(reference_string[i]);
47         page_faults++;
48         string_position++;
49     }
50 }
51
52 bool has_pages_faults(char *physical_memory, char c)
53 {
54     bool has = true;
55     for (int i = 0; i < NUMBER_OF_PAGES; i++)
56     {
57         if (physical_memory[i] == c)
58         {
59             has = false;
60         }
61     }
62     return has;
63 }

```

```
64
65 void refresh_memory(char *physical_memory, char new_char)
66 {
67     for (int i = 0; i < NUMBER_OF_PAGES; i++)
68     {
69         if (physical_memory[i] == fila.front())
70         {
71             physical_memory[i] = new_char;
72             fila.pop();
73             fila.push(new_char);
74             return;
75         }
76     }
77 }
78
79 void print_physical_memory(char *physical_memory)
80 {
81     for (int i = 0; i < NUMBER_OF_PAGES; i++)
82     {
83         cout << " " << physical_memory[i] << " ";
84     }
85 }
86
```