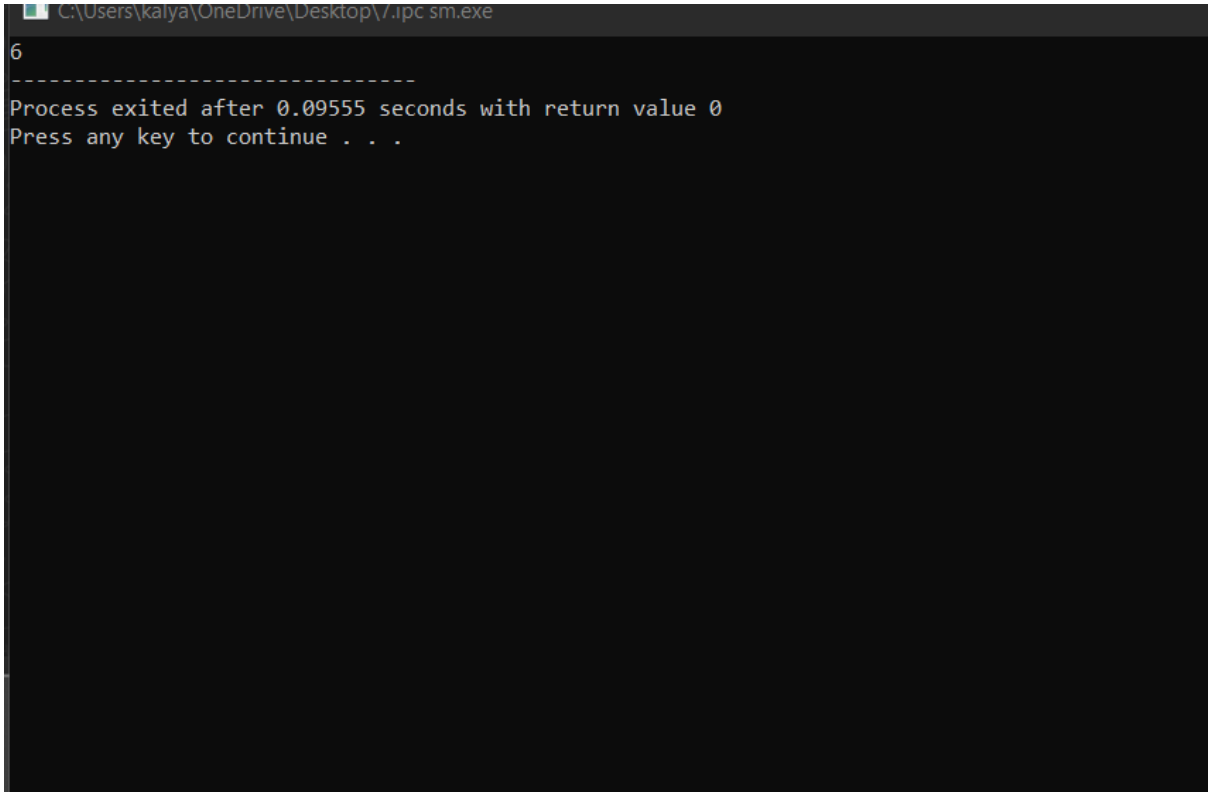
17. Construct a C program to simulate the Least Recently Used paging technique of memory management.

```
#include<bits/stdc++.h>
using namespace std;


int pageFaults(int pages[], int n, int capacity)
{

    unordered_set<int> s;


    unordered_map<int, int> indexes;


    int page_faults = 0;
    for (int i=0; i<n; i++)
    {

        if (s.size() < capacity)
        {
            if (s.find(pages[i])==s.end())
            {
                s.insert(pages[i]);


                page_faults++;
            }

            indexes[pages[i]] = i;
        }


        else
        {

            if (s.find(pages[i]) == s.end())
            {

                int lru = INT_MAX, val;
                for (auto it=s.begin(); it!=s.end(); it++)
                {
                    if (indexes[*it] < lru)
```

```
        {

            if (s.find(pages[i]) == s.end())
            {

                int lru = INT_MAX, val;
                for (auto it=s.begin(); it!=s.end(); it++)
                {
                    if (indexes[*it] < lru)
                    {
                        lru = indexes[*it];
                        val = *it;
                    }
                }


                s.erase(val);


                s.insert(pages[i]);


                page_faults++;
            }


            indexes[pages[i]] = i;
        }
    }

    return page_faults;
}

int main()
{
    int pages[] = {7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2};
    int n = sizeof(pages)/sizeof(pages[0]);
    int capacity = 4;
    cout << pageFaults(pages, n, capacity);
    return 0;
}
```

Output:


C:\Users\kalya\OneDrive\Desktop\7.ipc sm.exe
6

Process exited after 0.09555 seconds with return value 0
Press any key to continue . . .