OPERATING SYSTEMS

(CECSC09-I)

Submitted By:

Ashish Kumar

2019UCO1518

INDEX

S. No	TOPIC	Page no.
1	Write a program to create a child process and display the process id of the parent process from the child process.	3
2	Write a program to simulate any page replacement algorithm that suffers from Belady’s anomaly. Demonstrate the effect of Belady’s anomaly.	4
3	Write a program to simulate LOOK algorithm. Let the number of cylinders = 250.	7

Write a program to create a child process and display the process id of the parent process from the child process.

Code :

```c
#include <stdlib.h>
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
int main() {

    int id =  fork();
    if (id == 0)
    {
        printf("Hi! this is a child process with id : %d and whose parent process is : %d\n" , getpid() , getppid());
    }
    else
    {
        printf("Hi! this is a parent process with id : %d \n" , getpid());
    }
    return 0;
}
```

Output:

```
Hi! this is a child process with id : 20797 and whose parent process is : 20796
Hi! this is a parent process with id : 20796
```

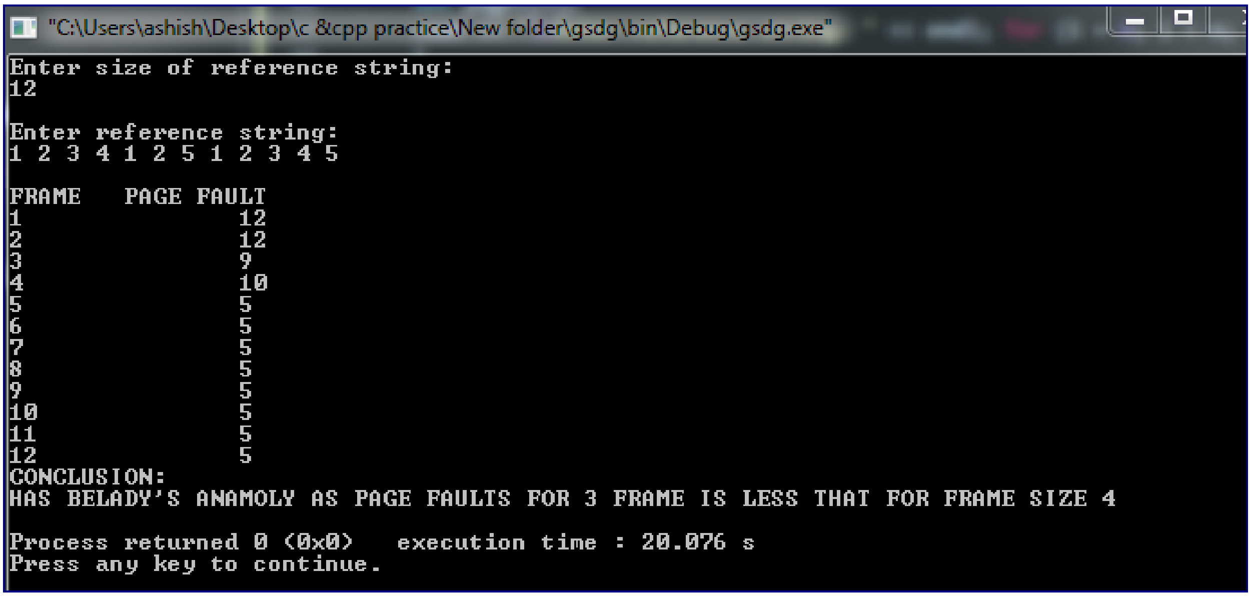
Write a program to simulate any page replacement algorithm that suffers from Belady’s anomaly. Demonstrate the effect of Belady’s anomaly.

Code :

```
#include <bits/stdc++.h>
using namespace std;
bool checkIsPresent(queue<int>frame, int x)
{
    while (!frame.empty())
    {
        if (frame.front() == x)
            return true;
        frame.pop();
    }
    return false;
}
int pageFaults(vector<int> & pages,  int capacity)
{   int n = pages.size();
    queue<int>frames;
    int total_faults = 0;
    for (int i = 0; i < n; i++)
    {
        if (frames.size() < capacity and !checkIsPresent(frames,
pages[i]))
            frames.push(pages[i]) , total_faults++;
        else
        {
            if (!checkIsPresent(frames, pages[i]))
            {
                frames.pop(); frames.push(pages[i]); total_faults
++;
```

```
            }
        }

    }
    return total_faults;
}

void solve()
{   // FCFS
    int i, n;
    cout << "Enter size of reference string: " << endl; cin >> n;
    vector<int> pages(n);
    cout << "\nEnter reference string: " << endl;
    for (i = 0; i < n; i++)
        cin >> pages[i];

    vector<int> pagefaults(n + 2);
    for (i = 1; i <= n; i++)
        pagefaults[i] = pageFaults(pages, i);

    cout << "\nFRAME\tPAGE FAULT\n";
    for (i = 1; i <= n; i++)
        cout << i << "\t\t" << pagefaults[i] << "\n";

    bool check = false;
    int fs, ls;
    for (i = 2; i <= n; i++)
    {
        if (pagefaults[i - 1] < pagefaults[i])
        {
            check = true , fs = i - 1 , ls = i; break;
        }
    }
    cout << "CONCLUSION: \n";
    if (!check)

        cout << "THERE IS NO BELADYS'S ANAMOLY IN THE GIVEN SEQUENCE \n";
    else
        cout << "HAS BELADY'S ANAMOLY AS PAGE FAULTS FOR " << fs << " FRAME IS LESS THAT FOR FRAME SIZE " << ls << "\n";
}

int main()
{

    solve();

    return 0;
}
```

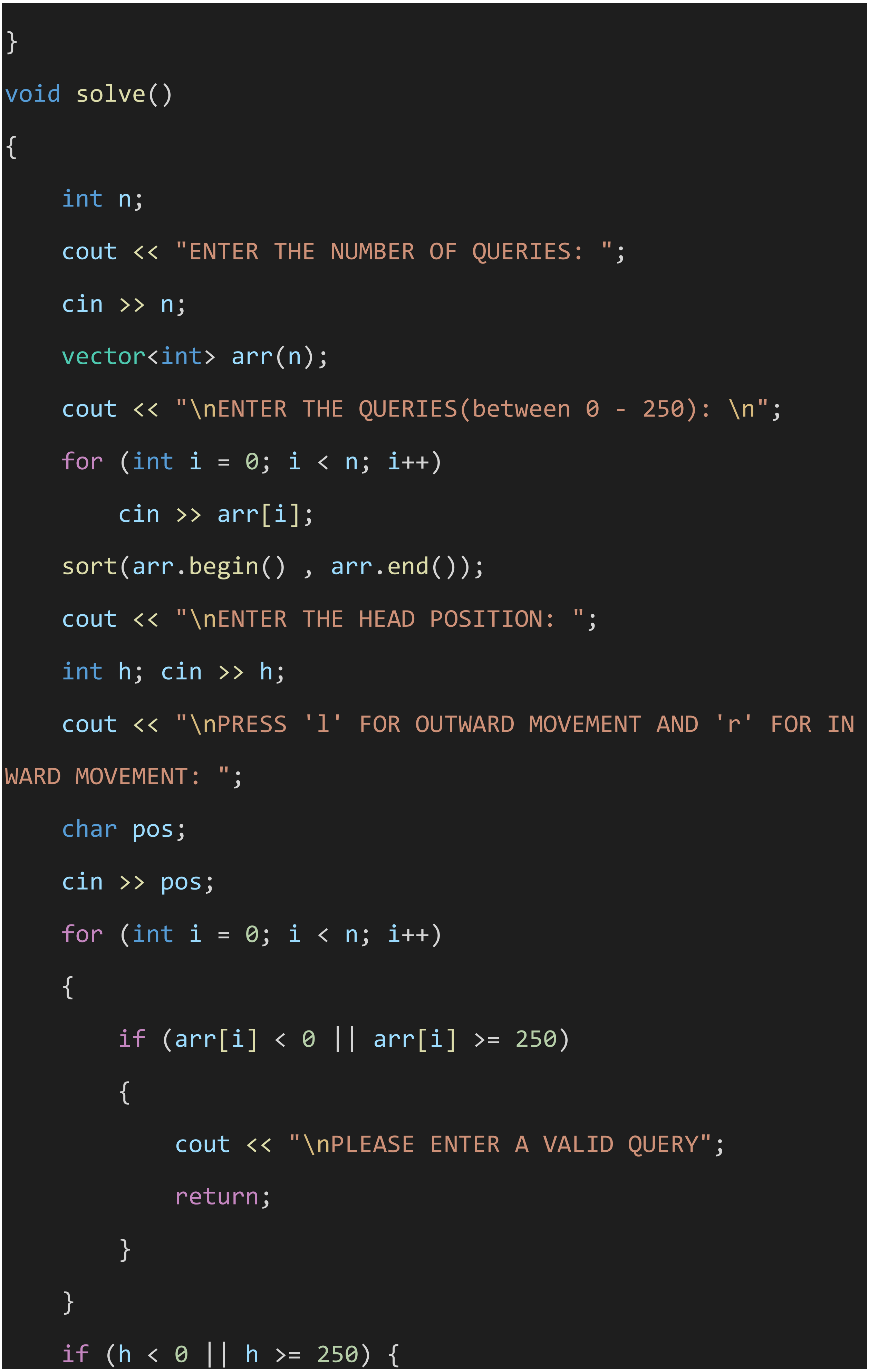
OUTPUT :


"C:\Users\ashish\Desktop\c &cpp practice\New folder\gsdg\bin\Debug\gsdg.exe"

```
Enter size of reference string:
12

Enter reference string:
1 2 3 4 1 2 5 1 2 3 4 5

FRAME    PAGE FAULT
1               12
2               12
3               9
4               10
5               5
6               5
7               5
8               5
9               5
10              5
11              5
12              5
CONCLUSION:
HAS BELADY'S ANAMOLY AS PAGE FAULTS FOR 3 FRAME IS LESS THAT FOR FRAME SIZE 4

Process returned 0 (0x0)   execution time : 20.076 s
Press any key to continue.
```

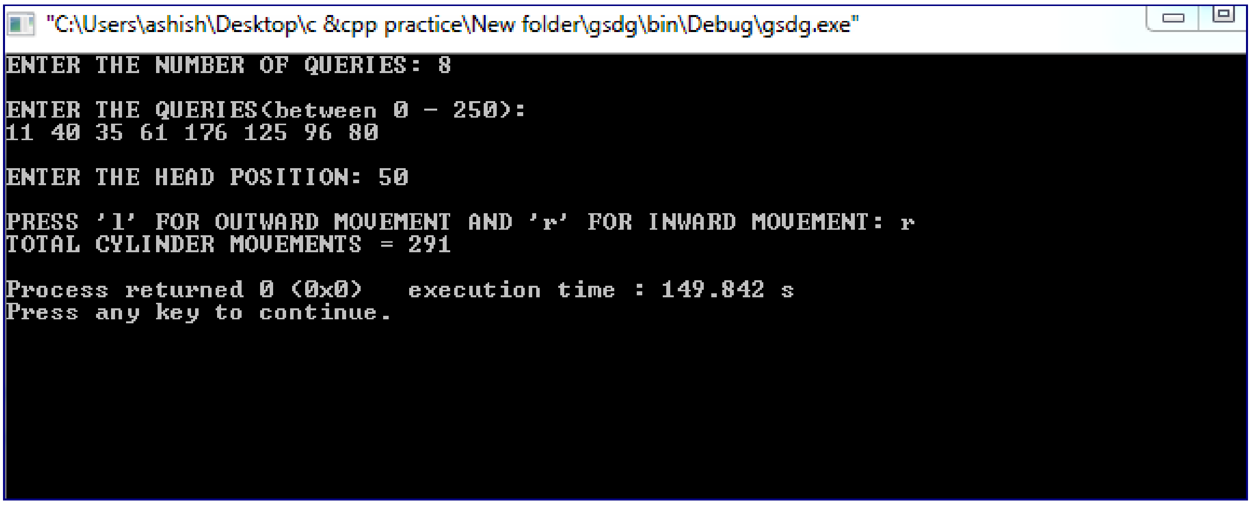
Write a program to simulate LOOK algorithm. Let the number of cylinders = 250.

```
#include<bits/stdc++.h>

using namespace std;

int look(vector<int>&arr, int h, char pos)

{   int n = arr.size();

    int ans = 0;

    int mx = arr[n - 1] , mn = arr[0];

    if (pos == 'l')

    {

        ans += max(h - arr[0], 0);

        if (h < arr[n - 1])

            ans += max(arr[n - 1] - arr[0], arr[n - 1] - h);

        return ans;

    }

    if (pos == 'r')

    {

        ans += max(arr[n - 1] - h, 0);

        if (h > arr[0])

        {

            ans += max(arr[n - 1] - arr[0], h - arr[0]);

        }

        return ans;

    }

    return -1;
```



```
}
void solve()
{
    int n;
    cout << "ENTER THE NUMBER OF QUERIES: ";
    cin >> n;
    vector<int> arr(n);
    cout << "\nENTER THE QUERIES(between 0 - 250): \n";
    for (int i = 0; i < n; i++)
        cin >> arr[i];
    sort(arr.begin() , arr.end());
    cout << "\nENTER THE HEAD POSITION: ";
    int h; cin >> h;
    cout << "\nPRESS 'l' FOR OUTWARD MOVEMENT AND 'r' FOR INWARD MOVEMENT: ";
    char pos;
    cin >> pos;
    for (int i = 0; i < n; i++)
    {
        if (arr[i] < 0 || arr[i] >= 250)
        {
            cout << "\nPLEASE ENTER A VALID QUERY";
            return;
        }
    }
    if (h < 0 || h >= 250) {
        cout << "\nPLEASE ENTER A VALID QUERY";
        return;
    }
    if (pos != 'l' && pos != 'r') {
        cout << "\nPLEASE ENTER A VALID QUERY";
        return;
    }
    cout << "TOTAL CYLINDER MOVEMENTS = " << look(arr, h, po
s)<< endl;
}

int main()
{
    solve();
    return 0;
}
```

Output :



"C:\Users\ashish\Desktop\c &cpp practice\New folder\gsdg\bin\Debug\gsdg.exe"

```
ENTER THE NUMBER OF QUERIES: 8

ENTER THE QUERIES(between 0 - 250):
11 40 35 61 176 125 96 80

ENTER THE HEAD POSITION: 50

PRESS 'l' FOR OUTWARD MOVEMENT AND 'r' FOR INWARD MOVEMENT: r
TOTAL CYLINDER MOVEMENTS = 291

Process returned 0 (0x0)   execution time : 149.842 s
Press any key to continue.
```