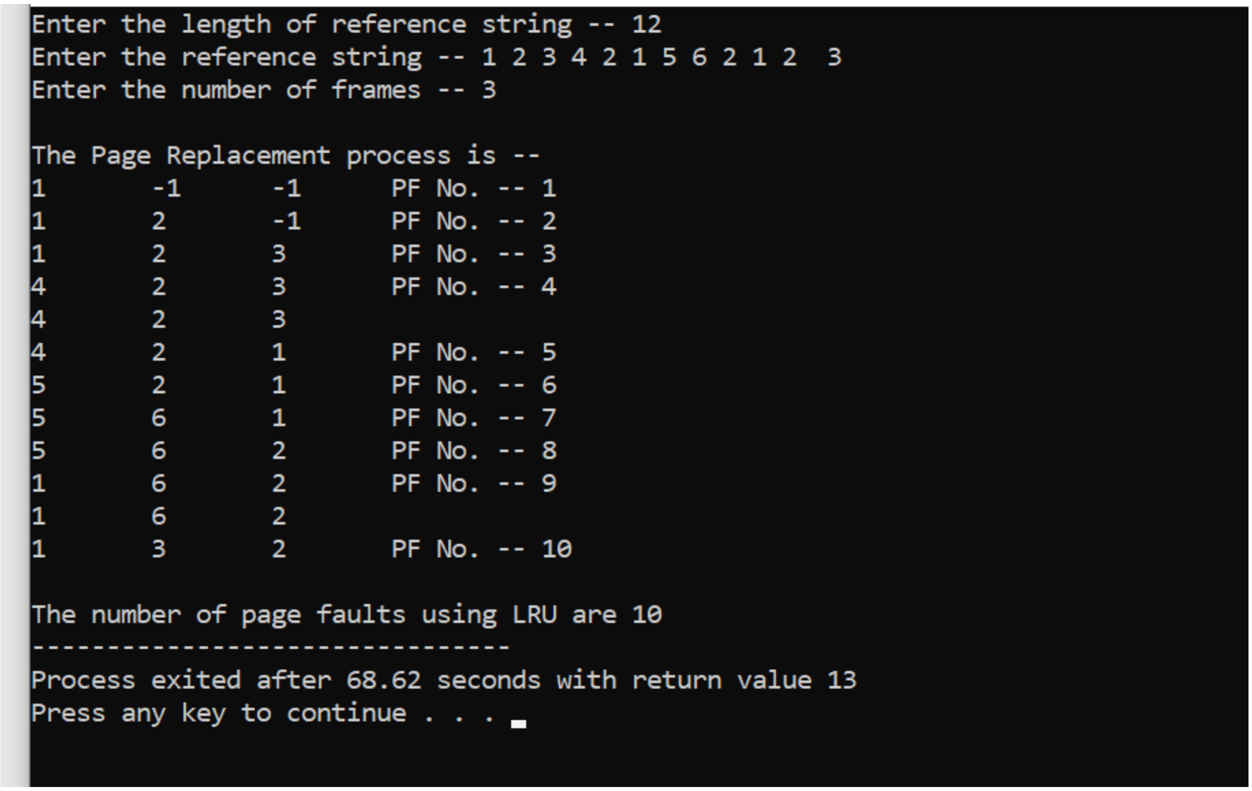
LAB # 12

Q) **Implement the above code and paste the screen shot of the output.**

LRU

```
#include<stdio.h>
#include<conio.h>
main()
{
int i, j , k, min, rs[25], m[10], count[10], flag[25], n, f, pf=0, next=1;
printf("Enter the length of reference string -- ");
scanf("%d",&n);
printf("Enter the reference string -- ");
for(i=0;i<n;i++)
{
scanf("%d",&rs[i]); flag[i]=0;
}
printf("Enter the number of frames -- ");
scanf("%d",&f); for(i=0;i<f;i++)
{
count[i]=0; m[i]=-1;
}
printf("\nThe Page Replacement process is -- \n");
for(i=0;i<n;i++)
{
for(j=0;j<f;j++)
{
```

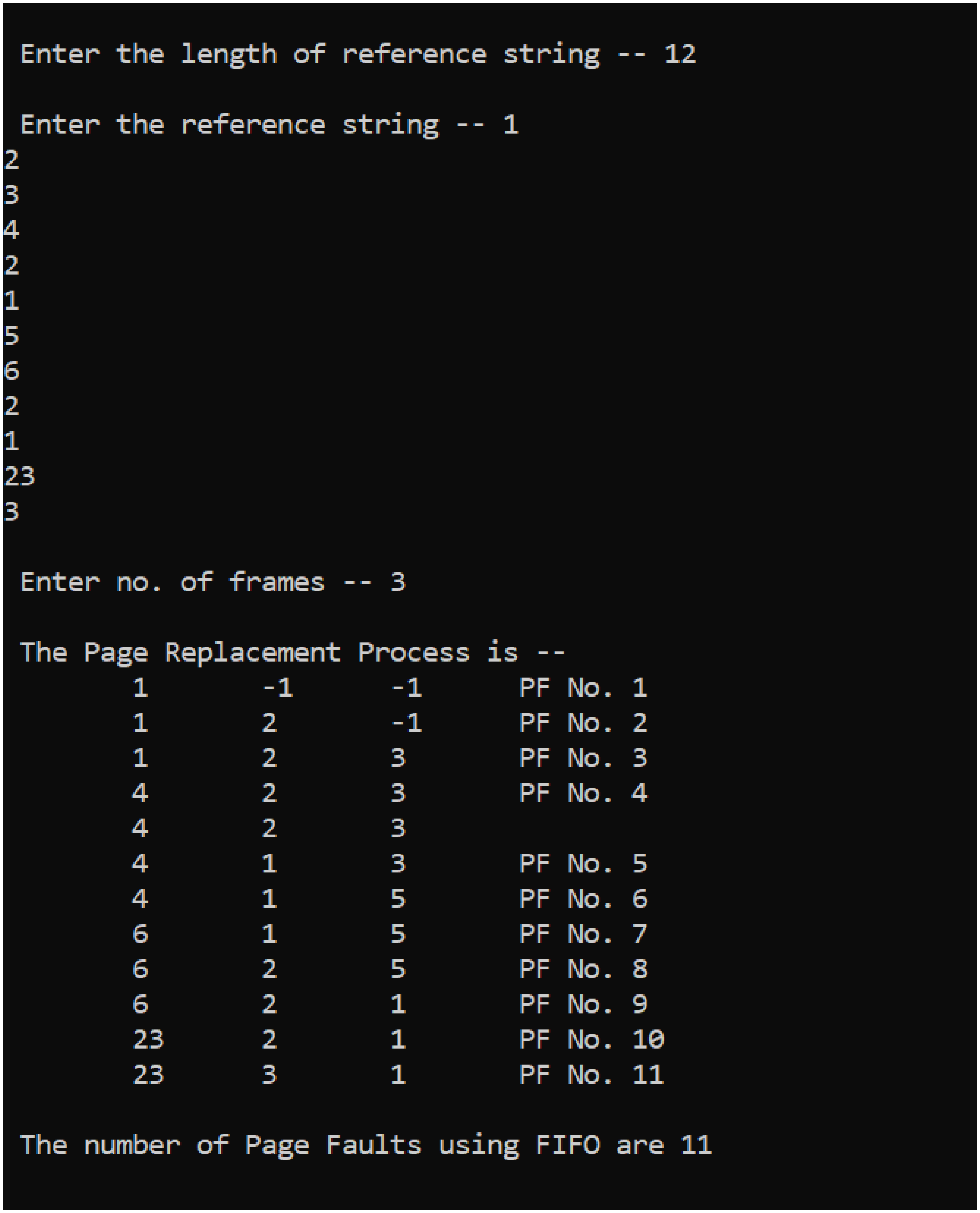
```
if(m[j]==rs[i])
{
flag[i]=1; count[j]=next;
next++;
}
}
if(flag[i]==0)
{
if(i<f)
{
m[i]=rs[i]; count[i]=next;
next++;
}
else
{
min=0;
for(j=1;j<f;j++) if(count[min] > count[j])
min=j;
m[min]=rs[i]; count[min]=next;
next++;
}

pf++;
}
for(j=0;j<f;j++)
printf("%d\t", m[j]);
if(flag[i]==0)
printf("PF No. -- %d" , pf);
printf("\n");
}
printf("\nThe number of page faults using LRU are %d",pf);
getch();
}
```



```
Enter the length of reference string -- 12
Enter the reference string -- 1 2 3 4 2 1 5 6 2 1 2  3
Enter the number of frames -- 3

The Page Replacement process is --
1       -1      -1      PF No. -- 1
1       2       -1      PF No. -- 2
1       2       3       PF No. -- 3
4       2       3       PF No. -- 4
4       2       3
4       2       1       PF No. -- 5
5       2       1       PF No. -- 6
5       6       1       PF No. -- 7
5       6       2       PF No. -- 8
1       6       2       PF No. -- 9
1       6       2
1       3       2       PF No. -- 10

The number of page faults using LRU are 10
--------------------------------
Process exited after 68.62 seconds with return value 13
Press any key to continue . . .
```

FIFO

```
#include<stdio.h>
#include<conio.h>
main()
{
int i, j, k, f, pf=0, count=0, rs[25], m[10], n;

printf("\n Enter the length of reference string -- ");
scanf("%d",&n);
printf("\n Enter the reference string -- ");
for(i=0;i<n;i++) scanf("%d",&rs[i]);
printf("\n Enter no. of frames -- ");
scanf("%d",&f); for(i=0;i<f;i++) m[i]=-1;
printf("\n The Page Replacement Process is -- \n");
for(i=0;i<n;i++)
{
for(k=0;k<f;k++)
{
if(m[k]==rs[i])
break;
}
if(k==f)
{
m[count++]=rs[i]; pf++;
}
for(j=0;j<f;j++)
printf("\t%d",m[j]);
if(k==f) printf("\tPF No. %d",pf);
printf("\n"); if(count==f) count=0;
}
printf("\n The number of Page Faults using FIFO are %d",pf);
getch();
}
```



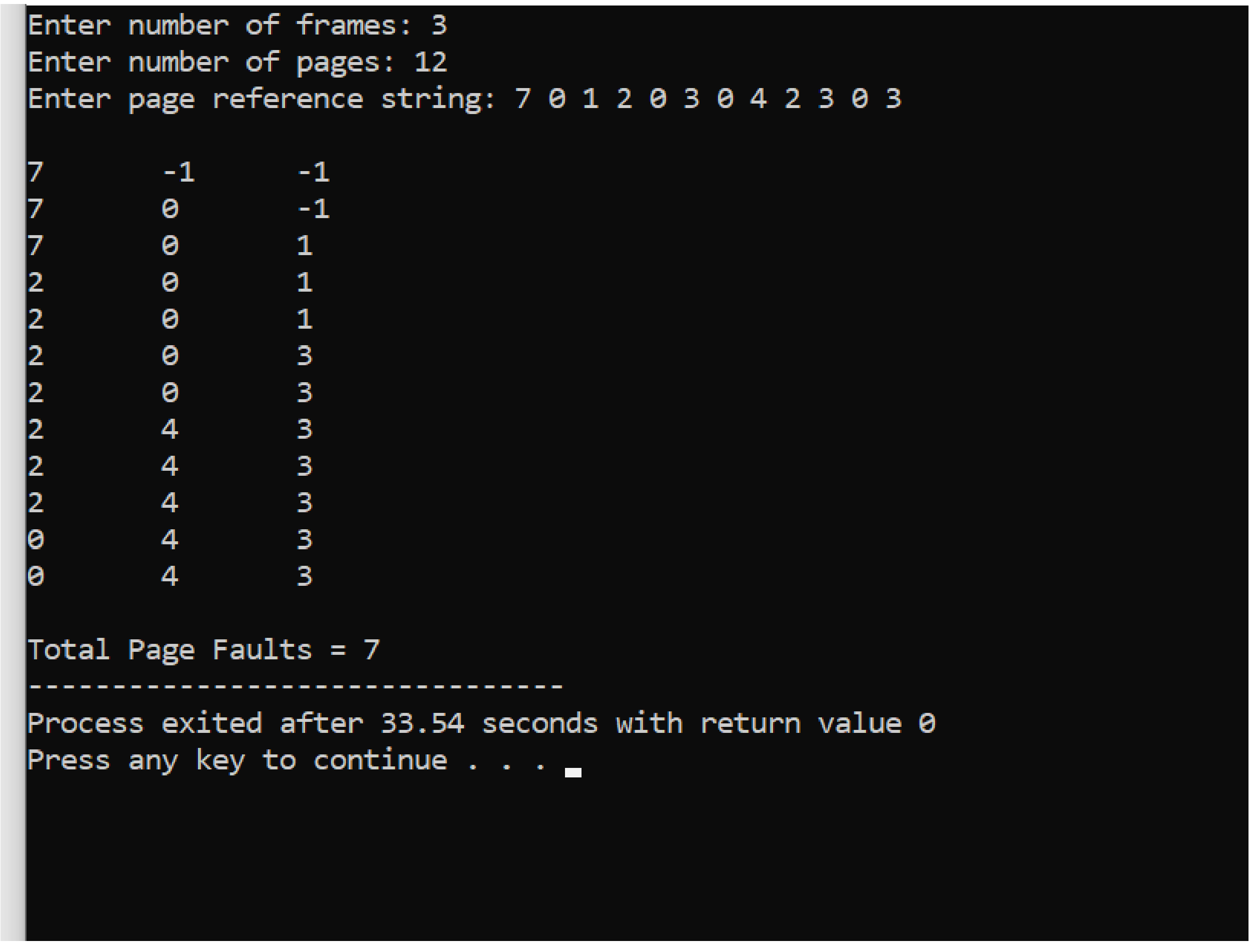
```
 Enter the length of reference string -- 12

 Enter the reference string -- 1
2
3
4
2
1
5
6
2
1
23
3

 Enter no. of frames -- 3

 The Page Replacement Process is --
        1       -1      -1      PF No. 1
        1       2       -1      PF No. 2
        1       2       3       PF No. 3
        4       2       3       PF No. 4
        4       2       3
        4       1       3       PF No. 5
        4       1       5       PF No. 6
        6       1       5       PF No. 7
        6       2       5       PF No. 8
        6       2       1       PF No. 9
        23      2       1       PF No. 10
        23      3       1       PF No. 11

 The number of Page Faults using FIFO are 11
```

OPTIMAL

```
#include <stdio.h>

int main() {
    int no_of_frames, no_of_pages, frames[10], pages[30];
    int temp[10], flag1, flag2, flag3, i, j, k, pos, max, faults = 0;

    printf("Enter number of frames: ");
    scanf("%d", &no_of_frames);

    printf("Enter number of pages: ");
    scanf("%d", &no_of_pages);

    printf("Enter page reference string: ");
    for (i = 0; i < no_of_pages; ++i) {
        scanf("%d", &pages[i]);
    }

    for (i = 0; i < no_of_frames; ++i) {
        frames[i] = -1;
    }

    for (i = 0; i < no_of_pages; ++i) {
        flag1 = flag2 = 0;

        for (j = 0; j < no_of_frames; ++j) {
            if (frames[j] == pages[i]) {
```

```
        flag1 = flag2 = 1;
        break;
    }
}

if (flag1 == 0) {
    for (j = 0; j < no_of_frames; ++j) {
        if (frames[j] == -1) {
            faults++;
            frames[j] = pages[i];
            flag2 = 1;
            break;
        }
    }
}

if (flag2 == 0) {
    flag3 = 0;

    for (j = 0; j < no_of_frames; ++j) {
        temp[j] = -1;
        for (k = i + 1; k < no_of_pages; ++k) {
            if (frames[j] == pages[k]) {
                temp[j] = k;
                break;
            }
        }
    }
```

```
            for (j = 0; j < no_of_frames; ++j) {
                if (temp[j] == -1) {
                    pos = j;
                    flag3 = 1;
                    break;
                }
            }

            if (flag3 == 0) {
                max = temp[0];
                pos = 0;

                for (j = 1; j < no_of_frames; ++j) {
                    if (temp[j] > max) {
                        max = temp[j];
                        pos = j;
                    }
                }
            }

            frames[pos] = pages[i];
            faults++;
        }

        printf("\n");

        for (j = 0; j < no_of_frames; ++j) {
            printf("%d\t", frames[j]);
        }
    }

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

    return 0;
}
```

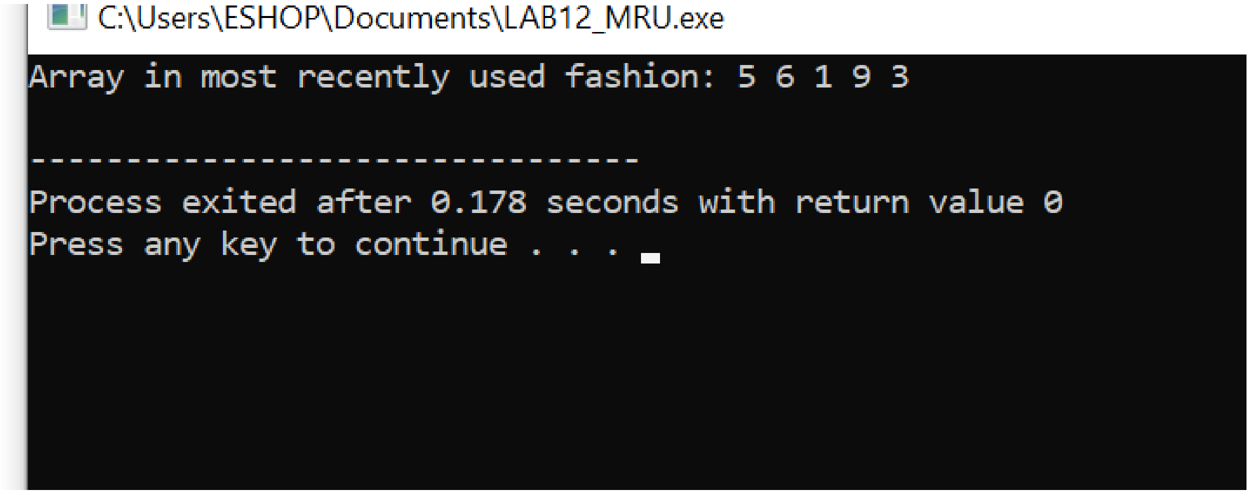


```
Enter number of frames: 3
Enter number of pages: 12
Enter page reference string: 7 0 1 2 0 3 0 4 2 3 0 3

7       -1      -1
7       0       -1
7       0       1
2       0       1
2       0       1
2       0       3
2       0       3
2       4       3
2       4       3
2       4       3
0       4       3
0       4       3

Total Page Faults = 7
--------------------------------
Process exited after 33.54 seconds with return value 0
Press any key to continue . . .
```

MRU

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

// Function to update the array in most recently used (MRU) fashion
void recently(int* arr, int size, int elem) {
           int index = 0;
    index = elem % size;
    int temp = index;
    int id = arr[index];

    while (temp > 0) {
        arr[temp] = arr[--temp];
    }

    arr[0] = id;
}

// Function to print array elements
void print(int* arr, int size) {
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}
```

```
int main() {
    int elem = 3;
    int arr[] = { 6, 1, 9, 5, 3 };
    int size = sizeof(arr) / sizeof(arr[0]);

    recently(arr, size, elem);

    cout << "Array in most recently used fashion: ";
    print(arr, size);

    return 0;
}
```



```
Array in most recently used fashion: 5 6 1 9 3

--------------------------------
Process exited after 0.178 seconds with return value 0
Press any key to continue . . .
```