7.Develop a C program to simulate page replacement algorithms: a) FIFO b) LRU

a) FIFO

PROGRAM

```
#include<stdio.h>
int fr[3];
void display() {
   int i;
   printf("\n");
   for (i = 0; i < 3; i++) {
      printf("%d\t", fr[i]);
   }
}

int main() {
   int i, j, page[12] = {2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2};
   int flag1 = 0, flag2 = 0, pf = 0, frsize = 3, top = 0;

   for (i = 0; i < 3; i++) {
      fr[i] = -1;
   }

   for (j = 0; j < 12; j++) {
      flag1 = 0;
      flag2 = 0;

      for (i = 0; i < frsize; i++) {
         if (fr[i] == page[j]) {
            flag1 = 1;
            flag2 = 1;
            break;
         }
      }

      if (flag1 == 0) {
         for (i = 0; i < frsize; i++) {
            if (fr[i] == -1) {
               fr[i] = page[j];
               flag2 = 1;
               pf++; // Increment page fault count for initial fills
               break;
            }
         }
      }

      if (flag2 == 0) {
         fr[top] = page[j];
         top++;
```

```
            pf++; // Increment page fault count for replacements
            if (top >= frsize) {
               top = 0;
            }
         }
         display();
      }

      printf("\nNumber of page faults : %d", pf);
      return 0;
}
```

OUTPUT:

```
2 -1 -1
2  3 -1
2  3 -1
2  3  1
5  3  1
5  2  1
5  2  4
5  2  4
3  2  4
3  2  4
3  5  4
3  5  2

Number of page faults: 9
```

b) LRU

PROGRAM

```
#include<stdio.h>

int fr[3];

void display() {
```

```
   int i;
   for (i = 0; i < 3; i++) {
      printf("\t%d", fr[i]);
   }
   printf("\n");
}

int findLRU(int time[], int n){
   int i, minimum = time[0], pos = 0;
   for(i = 1; i < n; ++i){
      if(time[i] < minimum){
         minimum = time[i];
         pos = i;
      }
   }
   return pos;
}

int main() {
   int p[12] = {2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2}, i, j;
   int fs[3], index, pf = 0, frsize = 3, counter = 0;
   int time[3];

   for(i = 0; i < frsize; i++) {
      fr[i] = -1;
   }

   for(j = 0; j < 12; j++) {
      int flag1 = 0, flag2 = 0;
      for(i = 0; i < frsize; i++) {
         if(fr[i] == p[j]) {
            counter++;
            time[i] = counter;
            flag1 = flag2 = 1;
            break;
         }
      }
      if(flag1 == 0) {
         for(i = 0; i < frsize; i++) {
            if(fr[i] == -1) {
               counter++;
               fr[i] = p[j];
               time[i] = counter;
               flag2 = 1;
               pf++;
               break;
            }
         }
      }
      if(flag2 == 0) {
```

```
            int pos = findLRU(time, frsize);
            counter++;
            fr[pos] = p[j];
            time[pos] = counter;
            pf++;
        }
        display();
    }

    printf("\nNumber of page faults: %d\n", pf);
    return 0;
}
```

OUTPUT:

```
2 -1 -1
2  3 -1
2  3 -1
2  3  1
2  5  1
2  5  1
2  5  4
2  5  4
3  5  4
3  5  2
3  5  2
3  5  2
```

No of page faults: 7

VIVA Questions:

1. How does FIFO differ from LRU in Handling Page Faults?

2. What is page replacement Algorithm?