CS266 LAB 11

NAME:

ARCHIT AGRAWAL

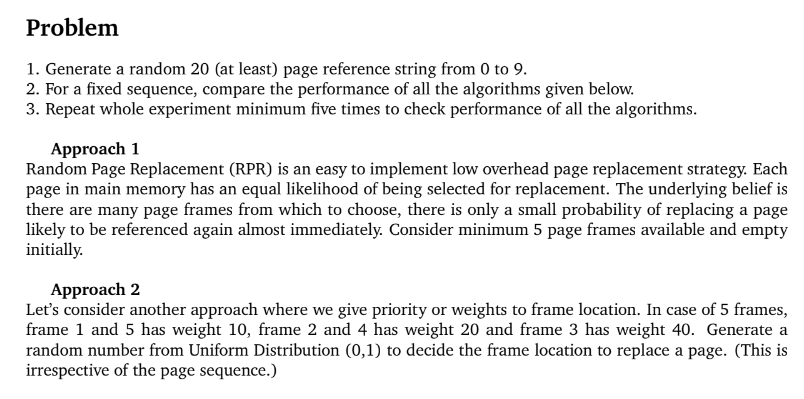
ROLL NO. :

202051213

SECTION:

2

***Question***



***Code***

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

void main(){

    int n = 5; //number of page frames, fixed

    int s; //length of referenceString

    printf("Enter the length of reference string (>= 20) : ");

    scanf("%d", &s);

    int referenceString[s];

    srand(time(0));

    printf("Reference String : ");

    for(int i = 0; i < s; i++){

        referenceString[i] = rand() % 10;

        printf("%d", referenceString[i]);

        if(i != s-1) printf(", ");

    }

    printf("\n");

    //same referenceString is passed to both the approaches

    printf("Page Faults in Equally Likely RPR (Approach 1) = %d\n", equallyLikelyRPR(s, referenceString, n));

    printf("Page Faults in Weighted RPR (Approach 2) = %d\n", weightedRPR(s, referenceString, n));

}

//Approach 1

int equallyLikelyRPR(int s, int referenceString[s], int n){

    int frames[n];

    int vacantFrames = 0; //stores the number of frames vacant at any moment

    for(int i = 0; i < n; i++) frames[i] = -1; //frames[i] == -1 indicates that frame is empty

    //frames[i] != -1 indicates that page number = frames[i] is stored at frames[i];

    int pageFaults = 0;

    for(int i = 0; i < s; i++){

        int flag = 0;   //a flag to check if the new page is already in the frames or not

        //if flag == 1, page is present already

        //else it is required to store it

        //this loop checks if new page is already present or not

        for(int j = 0; j < n; j++){

            if(frames[j] == referenceString[i]){

                flag = 1;

                break;

            }

        }

        //if new page is already present, there is no page fault, hence we can continue to next page

        if(flag == 1) continue;

        //if there are empty frames, and a page is not already present

        //we need to put this page in the empty space.

        //this will cause a page fault

        else if(vacantFrames < n){

            frames[vacantFrames] = referenceString[i];

            vacantFrames++;

            pageFaults++;

        }

        //else, if page is not present and no frames are vacant, we need to replace the page, and it will also cause a page fault

        else{

            int frameNum = rand() % 5; //each frame is equally likely to be replaced

            //printf("%d ", frameNum);

            frames[frameNum] = referenceString[i];

            pageFaults++;

        }

    }

    printf("\n");

    return pageFaults;

}

//Approach 2

int weightedRPR(int s, int referenceString[s], int n){

    int frames[n];

    int vacantFrames = 0;

    for(int i = 0; i < n; i++) frames[i] = -1; //frames[i] == -1 indicates that frame is empty

    //frames[i] != -1 indicates that page number = frames[i] is stored at frames[i];

    int pageFaults = 0;

    for(int i = 0; i < s; i++){

        int flag = 0;   //a flag to check if the new page is already in the frames or not

        //if flag == 1, page is present already

        //else it is required to store it

        //this loop checks if new page is already present or not

        for(int j = 0; j < n; j++){

            if(frames[j] == referenceString[i]){

                flag = 1;

                break;

            }

        }

        //if new page is already present, there is no page fault, hence we can continue to next page

        if(flag == 1) continue;

        //if there are empty frames, and a page is not already present

        //we need to put this page in the empty space.

        //this will cause a page fault

        else if(vacantFrames < n){

            frames[vacantFrames] = referenceString[i];

            vacantFrames++;

            pageFaults++;

        }

        //else, if page is not present and no frames are vacant, we need to replace the page, and it will also cause a page fault

        else{

            int randomNumber = rand() % 10;

            int frameNum = -1;

            //using switch case to create an distribution among the frame to be replaced

            //given frame 0 and 4 has 10% chances to be replaced

            //frame 1 and 3 has 20% chances to be replaced

            //frame 2 has 40% chances to be replaced

            //giving 1 number to frame 0 out of {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} generated by random function

            //will set its probability to be 10%.

            //Similarly giving 2 numbers (1, 2) to frame 1 will set its probability to be 20%;

            switch(randomNumber){

                case 0:

                    frameNum = 0;

                    break;

                case 1:

                case 2: //no break used, so randomNumber == 1 or randomNumber == 2, both points to frame 1

                    frameNum = 1;

                    break;

                case 3:

                case 4:

                case 5:

                case 6:  //no break used, so randomNumber in {3, 4, 5, 6} points to frame 2

                    frameNum = 2;

                    break;

                case 7:

                case 8: //no break used, so randomNumber == 7 or randomNumber == 8, both points to frame 3

                    frameNum = 3;

                    break;

                case 9:

                    frameNum = 4;

                    break;

                default:  //not really required

                    frameNum = -1;

            }

            //printf("%d ", frameNum);

            frames[frameNum] = referenceString[i];

            pageFaults++;

        }

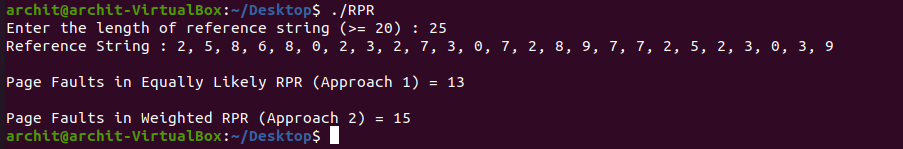
    }

    printf("\n");

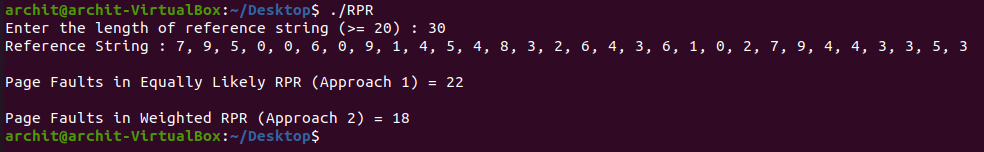
    return pageFaults;

}

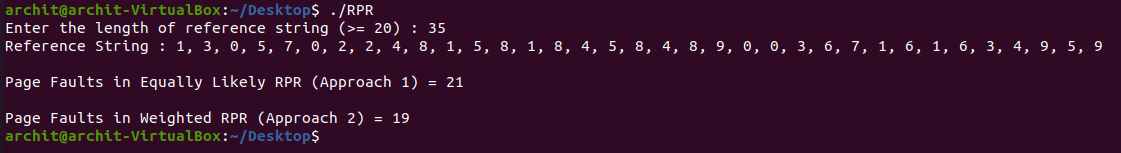
***Output 1***



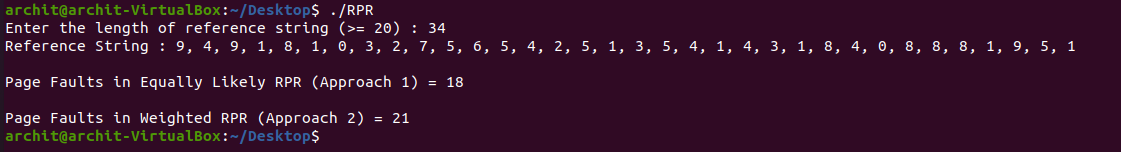
***Output 2***



***Output 3***



***Output 4***



***Output 5***

