***Aim: -*** To implement Suzuki-Kasami Algorithm.

***Program:-***

#include<stdio.h>

int max (int a, int b)

{

if (a>b)

return a;

else

return b;

}

int main()

{

int rn[5][5]={0};

int queue[10]={0};

int ln[5]={0};

int reqsite=0;

int i,j;

int sn=rn[0][0]+1;

rn[0][0]=rn[0][0]+1;

printf("the total no of process is 5");

printf("\nthe starting site is 0");

printf("\nsite 0 is requesting the token");

printf("\nsite 0 is broadcasting the message to all other sites\n\n");

for(i=0;i<5;i++)

{

rn[i][reqsite]=max(rn[i][reqsite],sn);

for(j=0;j<5;j++)

printf("\t%d",rn[i][j]);

printf("\n");

}

printf("\nas no site was having the token site 0 has acquired the token");

printf("\nsite 0 is executing the critical section");

printf("\nenter the site to enter the critical section");

scanf("%d",&reqsite);

sn=rn[reqsite][reqsite]+1;

for(i=0;i<5;i++)

{

rn[i][reqsite]=max(rn[i][reqsite],sn);

for(j=0;j<5;j++)

printf("\t%d",rn[i][j]);

printf("\n");

}

printf("\nthe token queue becomes\n\n");

queue[0]=reqsite;

for(i=0;i<10;i++)

{

printf("%d\t",queue[i]);

}

printf("\nreceiving the request message by site 0");

printf("\nsite 0 exiting the critical section");

ln[0]=rn[0][0];

printf("\nthe token is passed to site %d",reqsite);

return 0;

}

***Output :-***



