**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**FCFS SCHEDULING**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include<stdio.h>**

**struct process{**

**int name,bt,wt,tat;**

**}a[10];**

**int main(){**

**int n,i;**

**float awt=0.0,atat=0.0;**

**printf("Enter the number of processes : ");**

**scanf("%d",&n);**

**for(i=0;i<n;i++){**

**printf("Enter Burst time of the Process %d : ",i);**

**a[i].name=i;**

**scanf("%d",&a[i].bt);**

**}**

**a[0].wt=0;**

**for(i=1;i<n;i++)**

**a[i].wt=a[i-1].wt+a[i-1].bt;**

**for(i=0;i<n;i++){**

**a[i].tat=a[i].wt+a[i].bt;**

**awt+=a[i].wt;**

**atat+=a[i].tat;**

**}**

**awt=(float)awt/n;**

**atat=(float)atat/n;**

**printf("\n...................................................\n”);**

**printf(“FCFS SCHEDULING”);**

**printf("\n...................................................\n”);**

**printf("PROCESS\tBURST TIME\tWAITING TIME\tTURN AROUND TIME\n");**

**for(i=0;i<n;i++){**

**printf("P%d\t\t%d\t\t%d\t\t%d\n",a[i].name,a[i].bt,a[i].wt,a[i].tat);**

**}**

**printf("\nAverage Waiting Time(AWT) : %.2f\n",awt);**

**printf("Average Turn Around Time(ATAT) : %.2f\n",atat);**

**printf("\n.................\nGANTT CHART\n.................\n");**

**for(i=0;i<n;i++)**

**printf("| P%d ",a[i].name);**

**printf("|\n");**

**for(i=0;i<n;i++)**

**printf("%d ",a[i].wt);**

**printf(" %d\n",a[n-1].tat);**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Output**

**Enter the number of processes : 4**

**Enter Burst time of the Process 1 : 3**

**Enter Burst time of the Process 2 : 5**

**Enter Burst time of the Process 3 : 10**

**Enter Burst time of the Process 4 : 7**

**................................................................**

**FCFS SCHEDULING**

**................................................................**

**PROCESS BURST TIME WAITING TIME TURN AROUND TIME**

**P0 3 0 3**

**P1 5 3 8**

**P2 10 8 18**

**P3 7 18 25**

**Average Waiting Time(AWT) : 7.25**

**Average Turn Around Time(ATAT) : 13.50**

**| P0 | P1 | P2 | P3 |**

**0 3 8 18 25**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**SJF SCHEDULING**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include<stdio.h>**

**struct process{**

**int name,wt,tat,bt;**

**}a[20],temp;**

**int main(){**

**int n,i,j;**

**float awt=0.0,atat=0.0;**

**printf("Enter the number of processes : ");**

**scanf("%d",&n);**

**for(i=0;i<n;i++){**

**a[i].name=i;**

**printf("Enter Burst time of the Process %d : ",a[i].name);**

**scanf("%d",&a[i].bt);**

**}**

**for(i=0;i<n-1;i++)**

**for(j=0;j<n-i-1;j++)**

**if(a[j].bt>a[j+1].bt){**

**temp = a[j];**

**a[j] = a[j+1];**

**a[j+1] = temp;**

**}**

**a[0].wt=0;**

**for(i=1;i<n;i++)**

**a[i].wt=a[i-1].wt+a[i-1].bt;**

**for(i=0;i<n;i++){**

**a[i].tat=a[i].wt+a[i].bt;**

**}**

**for(i=0;i<n;i++){**

**awt+=a[i].wt;**

**atat+=a[i].tat;**

**}**

**atat=(float)atat/n;**

**awt=(float)awt/n;**

**printf("\n...................................................\n”);**

**printf(“SJF SCHEDULING”);**

**printf("\n...................................................\n”);**

**printf("PROCESS\tBURST TIME\tWAITING TIME\tTURN AROUND TIME\n");**

**for(i=0;i<n;i++){**

**printf("P%d\t\t%d\t\t%d\t\t%d\n",a[i].name,a[i].bt,a[i].wt,a[i].tat);**

**}**

**printf("\nAverage Waiting Time(AWT) : %.2f\n",awt);**

**printf("Average Turn Around Time(ATAT) : %.2f\n",atat);**

**printf("\n.................\nGANTT CHART\n.................\n");**

**for(i=0;i<n;i++)**

**printf("| P%d ",a[i].name);**

**printf("|\n");**

**for(i=0;i<n;i++)**

**printf("%d ",a[i].wt);**

**printf(" %d\n",a[n-1].tat);**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**OUTPUT**

**Enter the number of processes : 4**

**Enter Burst time of the Process 0 : 4**

**Enter Burst time of the Process 1 : 7**

**Enter Burst time of the Process 2 : 3**

**Enter Burst time of the Process 3 : 10**

**.................................................................**

**FCFS SCHEDULING**

**.................................................................**

**PROCESS BURST TIME WAITING TIME TURN AROUND TIME**

**P2 3 0 3**

**P0 4 3 7**

**P1 7 7 14**

**P3 10 14 24**

**Average Waiting Time(AWT) : 6.00**

**Average Turn Around Time(ATAT) : 12.00**

**….………………….……………………………………………………..**

**GANTT CHART**

**….……………………………………..…………………………………..**

**| P2 | P0 | P1 | P3 |**

**0 3 7 14 24**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name :ALLBIN THANKACHAN Roll no :66**

**PRIORITY SCHEDULING**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include<stdio.h>**

**struct process{**

**int name,wt,tat,bt,pr;**

**}a[20],temp;**

**int main(){**

**int n,i,j;**

**float awt=0.0,atat=0.0;**

**printf("Enter the number of processes : ");**

**scanf("%d",&n);**

**for(i=0;i<n;i++){**

**a[i].name=i;**

**printf("Enter Burst time of the Process %d : ",a[i].name);**

**scanf("%d",&a[i].bt);**

**printf("Enter Priority of the Process %d : ",a[i].name);**

**scanf("%d",&a[i].pr);**

**}**

**for(i=0;i<n-1;i++)**

**for(j=0;j<n-i-1;j++)**

**if(a[j].pr>a[j+1].pr){**

**temp = a[j];**

**a[j] = a[j+1];**

**a[j+1] = temp;**

**}**

**a[0].wt=0;**

**for(i=1;i<n;i++)**

**a[i].wt=a[i-1].wt+a[i-1].bt;**

**for(i=0;i<n;i++){**

**a[i].tat=a[i].wt+a[i].bt;**

**}**

**for(i=0;i<n;i++){**

**awt+=a[i].wt;**

**atat+=a[i].tat;**

**}**

**atat=(float)atat/n;**

**awt=(float)awt/n;**

**printf("\n...................................................\n”);**

**printf(“PRIORITY SCHEDULING”);**

**printf("\n...................................................\n”);**

**printf("PROCESS\tPRIORITY\tBURST TIME\tWAITING TIME\tTAT\n");**

**for(i=0;i<n;i++){**

**printf("P%d\t%d\t%d\t%d\t%d\n",a[i].name,a[i].pr,a[i].bt,a[i].wt,a[i].tat);**

**}**

**printf("\nAverage Waiting Time(AWT) : %.2f\n",awt);**

**printf("Average Turn Around Time(ATAT) : %.2f\n",atat);**

**printf("\n.................\nGANTTCHART\n.................\n");**

**for(i=0;i<n;i++)**

**printf("| P%d ",a[i].name);**

**printf("|\n");**

**for(i=0;i<n;i++)**

**printf("%d ",a[i].wt);**

**printf(" %d\n",a[n-1].tat);**

**}**

**OUTPUT**

**Enter the number of processes : 4**

**Enter Burst time of the Process 0 : 5**

**Enter Priority of the Process 0 : 2**

**Enter Burst time of the Process 1 : 4**

**Enter Priority of the Process 1 : 1**

**Enter Burst time of the Process 2 : 2**

**Enter Priority of the Process 2 : 3**

**Enter Burst time of the Process 3 : 6**

**Enter Priority of the Process 3 : 4**

**................................................................**

**PRIORITY SCHEDULING**

**................................................................**

**PROCESS PRIORITY BURST TIME WAITING TIME TAT**

**P1 1 4 0 4**

**P0 2 5 4 9**

**P2 3 2 9 11**

**P3 4 6 11 17**

**Average Waiting Time(AWT) : 6.00**

**Average Turn Around Time(ATAT) : 10.25**

**.................**

**GANTT CHART**

**.................**

**| P1 | P0 | P2 | P3 |**

**0 4 9 11 17**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**ROUND ROBIN SCHEDULING**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include<stdio.h>**

**struct process{**

**int name,wt,bt,tat,rt,at;**

**}a[20],temp;**

**int main(){**

**int n,num,tq,i,k,item,j=0,flag=0,sum=0,res[20];**

**float awt=0.0,atat=0.0;**

**printf("Enter the number of processes : ");**

**scanf("%d",&n);**

**num=n;**

**for(i=0;i<n;i++){**

**a[i].name=i;**

**printf("Enter Arrival Time of Process %d : ",i);**

**scanf("%d",&a[i].at);**

**printf("Enter Burst Time of Process %d : ",i);**

**scanf("%d",&a[i].bt);**

**a[i].rt=a[i].bt;**

**}**

**printf("Enter the time quantum : ");**

**scanf("%d",&tq);**

**for(i=0;i<n-1;i++)**

**for(k=0;k<n-i-1;k++)**

**if(a[k].at>a[k+1].at){**

**temp = a[k];**

**a[k] = a[k+1];**

**a[k+1] = temp;**

**}**

**printf("\n.................\nGANTTCHART\n.................\n");**

**for(i=0,sum=0;num!=0;){**

**if(a[i].rt<=tq && a[i].rt>0){**

**sum+=a[i].rt;**

**res[j]=sum;**

**j++;**

**printf("| P%d ",a[i].name);**

**a[i].rt=0;**

**flag=1;**

**}**

**else if(a[i].rt>0){**

**sum+=tq;**

**a[i].rt-=tq;**

**res[j]=sum;**

**j++;**

**printf("| P%d ",a[i].name);**

**}**

**if(a[i].rt == 0 && flag == 1){**

**num--;**

**a[i].wt = sum-a[i].bt-a[i].at;**

**a[i].tat = sum-a[i].at;**

**awt+=a[i].wt;**

**atat+=a[i].tat;**

**flag=0;**

**}**

**if(i==n-1)**

**i=0;**

**else if(a[i+1].at<=sum)**

**i++;**

**else i=0;**

**}**

**item=j;**

**printf("|\n");**

**printf("0 ");**

**for(j=0;j<item;j++){**

**printf("%d ",res[j]);**

**}**

**awt/=(float)n;**

**atat/=(float)n;**

**printf("\n...................................................\n”);**

**printf(“PRIORITY SCHEDULING”);**

**printf("\n...................................................\n”);**

**printf("PROCESS\tARRIVAL TIME\tBURST TIME\tWAITING TIME\tTAT\n");**

**for(i=0;i<n;i++)**

**printf("P%d\t%d\t%d\t%d\t %d\n",a[i].name,a[i].at,a[i].bt,a[i].wt,a[i].tat);**

**printf("Average Waiting Time (AWT) : %.2f\n",awt);**

**printf("Average Turn Around Time (ATAT) : %.2f\n",atat);**

**}**

**OUTPUT**

**Enter the number of processes : 4**

**Enter Arrival Time of Process 0 : 2**

**Enter Burst Time of Process 0 : 4**

**Enter Arrival Time of Process 1 : 0**

**Enter Burst Time of Process 1 : 8**

**Enter Arrival Time of Process 2 : 1**

**Enter Burst Time of Process 2 : 12**

**Enter Arrival Time of Process 3 : 3**

**Enter Burst Time of Process 3 : 10**

**Enter the time quantum : 4**

**............................................................**

**GANTT CHART**

**............................................................**

**| P1 | P2 | P0 | P3 | P1 | P2 | P3 | P2 | P3 |**

**0 4 8 12 16 20 24 28 32 34**

**............................................................**

**ROUND ROBIN SCHEDULING**

**............................................................**

**PROCESS ARRIVAL TIME BURST TIME WAITING TIME TAT**

**P1 0 8 12 20**

**P2 1 12 19 31**

**P0 2 4 6 10**

**P3 3 10 21 31**

**Average Waiting Time (AWT) : 14.50**

**Average Turn Around Time (ATAT) : 23.00**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**SYSTEM CALLS : opendir(),readdir(),closedir()**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include <stdio.h>**

**#include <unistd.h>**

**#include <stdlib.h>**

**#include <dirent.h>**

**int main(){**

**DIR \*d;**

**struct dirent \*dtr;**

**d=opendir(".");**

**while(dtr=readdir(d)){**

**printf("%s\n",dtr->d\_name);**

**}**

**closedir(d);**

**return 0;**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**OUTPUT**

**..**

**.**

**aa.out**

**fork.c**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**SYSTEM CALLS : fork(),getpid(),exit()**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include <stdio.h>**

**#include <unistd.h>**

**#include <stdlib.h>**

**#include <sys/wait.h>**

**int main(){**

**int pid=fork();**

**if(pid==0){**

**execlp("/bin/date","date",NULL);**

**exit(0);**

**}**

**else if (pid<0){**

**printf("Error");**

**exit(1);**

**}**

**else{**

**printf("PID of parent %d\n",getpid());**

**wait(NULL);**

**exit(0);**

**}}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**OUTPUT**

**PID of parent 5638**

**Fri Jun 23 06:48:41 PM UTC 2023**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**IPC USING SHARED MEMORY**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**//WRITER PROCESS:**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<sys/ipc.h>**

**#include<sys/shm.h>**

**void main(){**

**key\_t key=ftok("Hello",60);**

**int shmid=shmget(key,1024,0666|IPC\_CREAT);**

**char \*str=(char \*)shmat(shmid,(void\*)0,0);**

**printf("\n Enter your data : ");**

**scanf("%[^\n]", str);**

**printf("\n Data Written to Shared Memory ");**

**puts(str); shmdt(str); }**

**//READER PROCESS:**

**#include<stdio.h>**

**#include<sys/ipc.h>**

**#include<sys/shm.h>**

**void main(){**

**key\_t key=ftok("Hello",60);**

**int shmid=shmget(key,1024,0666 | IPC\_CREAT);**

**char \*str=(char\*)shmat(shmid,(void\*)0,0);**

**printf("\n Data read from Shared Memory: ");**

**puts(str);**

**shmdt(str);**

**shmctl(shmid,IPC\_RMID,NULL); }**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**OUTPUT**

**oslab:/home$ gcc write.c**

**oslab:/home$ ./a.out**

**Enter your data : ALLBIN THANKACHAN**

**Data Written to Shared Memory ALLBIN THANKACHAN**

**oslab:/home$ gcc read.c**

**oslab:/home$ ./a.out**

**Data read from Shared Memory: ALLBIN THANKACHAN**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**PRODUCER CONSUMER USING SEMAPHORE**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include <stdio.h>**

**#include <stdlib.h>**

**int mutex = 1, full = 0;**

**int empty = 5, x = 0;**

**void producer(){**

**--mutex;**

**++full;**

**--empty;x++;**

**printf("\nProducer produces item %d",x);**

**++mutex;**

**}**

**void consumer(){**

**--mutex;**

**--full;**

**++empty;**

**printf("\nConsumer consumes item %d",x);**

**x--;**

**++mutex;**

**}**

**int main(){**

**int n, i;**

**printf("Buffer size = 5\n");**

**printf("1.Producer 2.Consumer 3.Exit");**

**for (i = 1; i > 0; i++){**

**printf("nEnter your choice:");**

**scanf("%d", &n);**

**switch (n){**

**case 1: if ((mutex == 1)&&(empty != 0))**

**producer();**

**else printf("Buffer is full!");**

**break;**

**case 2 : if ((mutex == 1)&&(full != 0))**

**consumer();**

**else printf("Buffer is empty!");**

**break;**

**case 3 : exit(0);**

**break;**

**}}}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Output**

**1.Producer 2.Consumer 3.Exit**

**Enter your choice:1**

**Producer produces item 1**

**Enter your choice:1**

**Producer produces item 2**

**Enter your choice:2**

**Consumer consumes item 2**

**Enter your choice:2**

**Consumer consumes item 1**

**Enter your choice:2**

**Buffer is empty!**

**Enter your choice:3**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**BANKER'S ALGORITHM**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include<stdio.h>**

**int main(){**

**int n,m,y,i,j,k,num=0,flag;**

**printf("Enter the number of processes : ");**

**scanf("%d",&n);**

**printf("Enter the number of resources : ");**

**scanf("%d",&m);**

**int max[n][m],alloc[n][m],need[n][m],avail[m];**

**int av[n][m],f[n],p[n];**

**printf("Enter the max resource of each process");**

**for(i=0;i<n;i++){**

**printf("\nMax Resource Allocation for process %d : ",i);**

**for(j=0;j<m;j++)**

**scanf("%d",&max[i][j]);**

**}**

**printf("Enter the allocated resource of each process");**

**for(i=0;i<n;i++){**

**printf("\nAllocated Resource Allocation for process %d : ",i);**

**for(j=0;j<m;j++)**

**scanf("%d",&alloc[i][j]);**

**}**

**printf("\nEnter the resource available to Process 0 : ");**

**for(j=0;j<m;j++)**

**scanf("%d",&avail[j]);**

**for(k=0;k<n;k++)**

**f[k]=0;**

**for(i=0;i<n;i++)**

**for(j=0;j<m;j++)**

**need[i][j]=max[i][j]-alloc[i][j];**

**for(k=0;k<n;k++){**

**for(i=0;i<n;i++){**

**if(f[i]==0){**

**flag=0;**

**for(j=0;j<m;j++)**

**if(need[i][j]>avail[j]){**

**flag=1;**

**break;**

**}**

**if(flag==0){**

**p[num]=i;**

**num++;**

**for(y=0;y<m;y++){**

**av[i][y]=avail[y];**

**avail[y]+=alloc[i][y];**

**}**

**f[i]=1;**

**}}}}**

**flag=1;**

**for(i=0;i<n;i++){**

**if(f[i]==0){**

**flag=0;**

**printf("The following System is not Safe!\n");**

**break;**

**}}**

**if(flag==1){**

**printf("System is Safe in the following order : \n");**

**for(i=0;i<n-1;i++)**

**printf("P%d -> ",p[i]);**

**printf("P%d ",p[n-1]);**

**}**

**printf("\n.................................................\n”);**

**printf(“FCFS SCHEDULING”);**

**printf("\n...................................................\n”);**

**printf("PROCESS\tMAX\tALLOC\tAVAIL\tNEED\n");**

**for(i=0;i<n;i++){**

**printf("P%d\t",i);**

**for(j=0;j<m;j++)**

**printf("%d",max[i][j]);**

**printf("\t");**

**for(j=0;j<m;j++)**

**printf("%d",alloc[i][j]);**

**printf("\t");**

**for(j=0;j<m;j++)**

**printf("%d",av[i][j]);**

**printf("\t");**

**for(j=0;j<m;j++)**

**printf("%d",need[i][j]);**

**printf("\n");**

**}}**

**OUTPUT**

**Enter the number of processes : 5**

**Enter the number of resources : 3**

**Enter the max resource of each process**

**Max Resource Allocation for process 0 : 7 5 3**

**Max Resource Allocation for process 1 : 3 2 2**

**Max Resource Allocation for process 2 : 9 0 2**

**Max Resource Allocation for process 3 : 2 2 2**

**Max Resource Allocation for process 4 : 4 3 3**

**Enter the allocated resource of each process**

**Allocated Resource Allocation for process 0 : 0 1 0**

**Allocated Resource Allocation for process 1 : 2 0 0**

**Allocated Resource Allocation for process 2 : 3 0 2**

**Allocated Resource Allocation for process 3 : 2 1 1**

**Allocated Resource Allocation for process 4 : 0 0 2**

**Enter the resource available to Process 0 : 3 3 2**

**System is Safe in the following order :**

**P1 -> P3 -> P4 -> P0 -> P2**

**.............................................**

**BANKER'S ALGORITHM**

**.............................................**

**PROCESS MAX ALLOC AVAIL NEED**

**P0 753 010 745 743**

**P1 322 200 332 122**

**P2 902 302 755 600**

**P3 222 211 532 011**

**P4 433 002 743 431**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**Page Replacement-FIFO**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include<stdio.h>**

**int main(){**

**int i,j,n,f,a[10],pg[10],flag,c=0,k=0;**

**printf("Enter the number of pages : ");**

**scanf("%d",&n);**

**printf("Enter the pages : ");**

**for(i=0;i<n;i++)**

**scanf("%d",&a[i]);**

**printf("Enter the number of frames : ");**

**scanf("%d",&f);**

**for(i=0;i<f;i++)**

**pg[i]=-1;**

**printf("FIFO Page Replacement\n..........................\n");**

**printf("REFERENCE STRING\tPAGE NUMBER\tSTATUS\n");**

**for(i=0;i<n;i++){**

**flag=0;**

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

**for(j=0;j<f;j++)**

**if(pg[j]==a[i])**

**flag=1;**

**if(flag==0){**

**for(k=0;k<j;k++)**

**pg[k]=pg[k+1];**

**pg[f-1]=a[i];**

**c++;**

**}**

**for(j=0;j<f;j++){**

**if(pg[j]!=-1)**

**printf("%d ",pg[j]);**

**}**

**if(flag==1)**

**printf("\t\t HIT\n");**

**else printf("\t\t MISS\n");**

**}**

**printf("Page Fault : %d\n",c);**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**OUTPUT**

**Enter the number of pages : 5**

**Enter the pages : 1**

**3**

**0**

**3**

**5**

**Enter the number of frames : 3**

**FIFO Page Replacement**

**..........................**

**REFERENCE STRING PAGE NUMBER STATUS**

**1 1 MISS**

**3 1 3 MISS**

**0 1 3 0 MISS**

**3 1 3 0 HIT**

**5 3 0 5 MISS**

**Page Fault : 4**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**DISK SHEDULING : FCFS,SCAN,CSCAN**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include<stdio.h>**

**void fcfs(int m, int a[], int hd);**

**void scan(int m, int a[], int hd);**

**void cscan(int m, int a[], int hd);**

**int main(){**

**int r[100], i, n, h;**

**printf("\nEnter the number of requests: ");**

**scanf("%d", &n);**

**printf("\nEnter the requesting sequence: ");**

**for (i = 0; i < n; i++)**

**scanf("%d", &r[i]);**

**printf("\nEnter the current head position: ");**

**scanf("%d", &h);**

**fcfs(n, r, h);**

**scan(n, r, h);**

**cscan(n, r, h);**

**return 0;**

**}**

**void fcfs(int m, int a[], int hd){**

**printf("\nFIRST COME FIRST SERVED\n");**

**int tst = 0, i, x;**

**for (i = 0; i < m; i++){**

**x = a[i] - hd;**

**if (x < 0)**

**x = -x;**

**tst = tst + x;**

**hd = a[i];**

**}**

**printf("\nScheduled sequence is:\n");**

**for (i = 0; i < m; i++)**

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

**printf("\nTotal seek time is %d.\n", tst);**

**}**

**void scan(int m, int a[], int hd){**

**printf("\nSCAN...\n");**

**int tst = 0, i, j, x, size, mov, temp, index;**

**printf("\nEnter the total disk size: ");**

**scanf("%d", &size);**

**size = size - 1;**

**printf("\nEnter the head movement direction (1 for high,**

**0 for low): ");**

**scanf("%d", &mov);**

**for (i = 0; i < m - 1; i++){**

**for (j=0;j<m-i-1;j++){**

**if(a[j]>a[j+1]){**

**temp = a[j];**

**a[j] = a[j + 1];**

**a[j + 1] = temp;**

**}}}**

**index = i;**

**for (i = 0; i < m; i++){**

**if (hd <= a[i]){**

**index = i;**

**break;**

**}}**

**if (mov == 1){**

**tst = (size - hd) + (size - a[0]);**

**printf("\nScheduled sequence is:\n");**

**for (i = index; i < m; i++)**

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

**for (i = index - 1; i >= 0; i--)**

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

**printf("\nTotal seek time is %d.\n", tst);**

**}**

**else{**

**tst = (hd - a[0]) + (size - a[0]);**

**printf("\nScheduled sequence is:\n");**

**for (i = index - 1; i >= 0; i--)**

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

**for (i = index; i < m; i++)**

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

**printf("\nTotal seek time is %d.\n", tst);**

**}}**

**void cscan(int m, int a[], int hd){**

**printf("\nC-SCAN\n");**

**int tst = 0, i, j, x, size, mov, temp, index;**

**printf("\nEnter the total disk size: ");**

**scanf("%d", &size);**

**size = size - 1;**

**printf("\nEnter the head movement direction (1 for high,**

**0 for low): ");**

**scanf("%d", &mov);**

**for (i = 0; i < m - 1; i++){**

**for (j = 0; j < m - i - 1; j++){**

**if (a[j] > a[j + 1]){**

**temp = a[j];**

**a[j] = a[j + 1];**

**a[j + 1] = temp;**

**}}}**

**index = i;**

**for (i = 0; i < m; i++){**

**if (hd <= a[i]){**

**index = i;**

**break;**

**}}**

**if (mov == 1){**

**tst = (size - hd) + (size - 0) + (a[index - 1] - 0);**

**printf("\nScheduled sequence is:\n");**

**for (i = index; i < m; i++)**

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

**for (i = 0; i < index - 1; i++)**

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

**printf("\nTotal seek time is %d.\n", tst);**

**}**

**else{**

**tst = hd + size;**

**printf("\nScheduled sequence is:\n");**

**for (i = index - 1; i >= 0; i--)**

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

**for (i = index; i < m; i++)**

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

**printf("\nTotal seek time is %d.\n", tst);**

**}}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**OUTPUT**

**Enter the number of requests: 7**

**Enter the requesting sequence: 82 170 43 140 24 16 190**

**Enter the current head position: 50**

**FCFS**

**Scheduled sequence is:**

**82 170 43 140 24 16 190**

**Total seek time is 642.**

**SCAN**

**Enter the total disk size: 199**

**Enter the head movement direction (1 for high, 0 for low): 1**

**Scheduled sequence is:**

**82 140 170 190 43 24 16**

**Total seek time is 330.**

**C-SCAN**

**Enter the total disk size: 199**

**Enter the head movement direction (1 for high, 0 for low): 1**

**Scheduled sequence is:**

**82 140 170 190 16 24**

**Total seek time is 389.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**FIRST FIT**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include<stdio.h>**

**void firstFit(int memory[],int m,int data[],int n){**

**int i, j,remain[10],flag=0;**

**int allocation[n];**

**for(i = 0; i < n; i++){**

**allocation[i] = -1;**

**}**

**for (i = 0; i < n; i++){**

**for (j = 0; j < m; j++){**

**if (memory[j] >= data[i]){**

**allocation[i] = j;**

**memory[j] -= data[i];**

**remain[i]=memory[j];**

**break;**

**}}}**

**printf("\nProcess No.\tProcess Size\tBlock no.\tRemain\n");**

**for(i=0;i<n;i++){**

**printf(" %i\t\t\t", i+1);**

**printf("%i\t\t", data[i]);**

**if (allocation[i] != -1)**

**printf("%i\t", allocation[i] + 1);**

**else{**

**flag=1;**

**printf("Not Allocated");**

**}if(flag==0)**

**printf("%d",remain[i]);**

**else printf("-");**

**printf("\n");**

**}}**

**int main(){**

**int m,n,i,data[10],memory[10];**

**printf("\nEnter The Number of Memory Blocks: ");**

**scanf("%d",&m);**

**for(i=0;i<m;i++){**

**printf("Enter size of block %d : ",i+1);**

**scanf("%d",&memory[i]);**

**}**

**printf("\nEnter The Number of data to insert : ");**

**scanf("%d",&n);**

**for(i=0;i<n;i++){**

**printf("Enter size of data %d : ",i+1);**

**scanf("%d",&data[i]);**

**}**

**firstFit(memory, m, data, n);**

**return 0 ;**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**OUTPUT**

**Enter The Number of Memory Blocks: 5**

**Enter size of block 1 : 100**

**Enter size of block 2 : 500**

**Enter size of block 3 : 200**

**Enter size of block 4 : 300**

**Enter size of block 5 : 600**

**Enter The Number of data to insert : 4**

**Enter size of data 1 : 212**

**Enter size of data 2 : 417**

**Enter size of data 3 : 112**

**Enter size of data 4 : 426**

**Process No. Process Size Block no. Remain**

**1 212 2 288**

**2 417 5 183**

**3 112 2 176**

**4 426 Not Allocated -**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**BEST FIT**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include <stdio.h>**

**void bestFit(int memory[],int m,int data[],int n){**

**int allocation[n];**

**int occupied[m];**

**int i,j;**

**for(i=0;i<n;i++){**

**allocation[i] = -1; }**

**for(i=0;i<m;i++){**

**occupied[i] = 0; }**

**for (i=0;i<n;i++){**

**int indexPlaced = -1;**

**for (j=0;j<m;j++){**

**if (memory[j]>=data[i] && !occupied[j]){**

**if (indexPlaced == -1)**

**indexPlaced = j;**

**else if (memory[j] < memory[indexPlaced])**

**indexPlaced = j; }}**

**if (indexPlaced!=-1){**

**allocation[i] = indexPlaced;**

**occupied[indexPlaced] = 1; }}**

**printf("\nProcess No.\tProcess Size\tBlock no.\n");**

**for(i=0;i<n;i++){**

**printf("%d \t\t\t %d \t\t\t", i+1, data[i]);**

**if (allocation[i] != -1)**

**printf("%d\n",allocation[i] + 1);**

**else printf("Not Allocated\n");**

**}}**

**int main(){**

**int m,n,i,data[10],memory[10];**

**printf("\nEnter The Number of Memory m: ");**

**scanf("%d",&m);**

**for(i=0;i<m;i++){**

**printf("Enter size of block %d : ",i+1);**

**scanf("%d",&memory[i]); }**

**printf("\nEnter The Number of data to insert : ");**

**scanf("%d",&n);**

**for(i=0;i<n;i++){**

**printf("Enter size of data %d : ",i+1);**

**scanf("%d",&data[i]); }**

**bestFit(memory, m, data, n);**

**return 0 ;**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**OUTPUT**

**Enter The Number of Memory m: 5**

**Enter size of block 1 : 100**

**Enter size of block 2 : 50**

**Enter size of block 3 : 30**

**Enter size of block 4 : 120**

**Enter size of block 5 : 35**

**Enter The Number of data to insert : 4**

**Enter size of data 1 : 40**

**Enter size of data 2 : 10**

**Enter size of data 3 : 30**

**Enter size of data 4 : 60**

**Process No. Process Size Block no.**

**1 40 2**

**2 10 3**

**3 30 5**

**4 60 1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name : ALLBIN THANKACHAN Roll no :66**

**WORST FIT**

**Date : Exp no :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**#include <stdio.h>**

**void worstFit(int memory[], int m, int data[], int n){**

**int allocation[n];**

**for(int i = 0; i < n; i++){**

**allocation[i] = -1;**

**}**

**for (int i=0; i<n; i++){**

**int indexPlaced = -1;**

**for (int j=0; j<m; j++){**

**if (memory[j] >= data[i]) {**

**if (indexPlaced == -1)**

**indexPlaced = j;**

**else if (memory[indexPlaced] < memory[j])**

**indexPlaced = j;**

**}}**

**if (indexPlaced != -1){**

**allocation[i] = indexPlaced;**

**memory[indexPlaced] -= data[i];**

**}}**

**printf("\nProcess No.\tProcess Size\tBlock no.\n");**

**for (int i = 0; i < n; i++){**

**printf("%d \t\t\t %d \t\t\t", i+1, data[i]);**

**if (allocation[i] != -1)**

**printf("%d\n",allocation[i] + 1);**

**else printf("Not Allocated\n");**

**}}**

**int main(){**

**int m,n,i,data[10],memory[10];**

**printf("\nEnter the number of Memory blocks : ");**

**scanf("%d",&m);**

**for(i=0;i<m;i++){**

**printf("Enter size of block %d : ",i+1);**

**scanf("%d",&memory[i]);**

**}**

**printf("\nEnter The Number of data to insert : ");**

**scanf("%d",&n);**

**for(i=0;i<n;i++){**

**printf("Enter size of data %d : ",i+1);**

**scanf("%d",&data[i]);**

**}worstFit(memory, m, data, n);**

**return 0 ;**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**OUTPUT**

**Enter the number of Memory blocks : 5**

**Enter size of block 1 : 5**

**Enter size of block 2 : 4**

**Enter size of block 3 : 3**

**Enter size of block 4 : 6**

**Enter size of block 5 : 7**

**Enter The Number of data to insert : 4**

**Enter size of data 1 : 1**

**Enter size of data 2 : 3**

**Enter size of data 3 : 5**

**Enter size of data 4 : 3**

**Process No. Process Size Block no.**

**1 1 5**

**2 3 4**

**3 5 5**

**4 3 1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***