**Practical : 11**

**Aim: Implementation of Worst-Fit Memory Allocation Algorithm**

**Program:**

**#include <stdio.h>**

**typedef struct**

**{**

**int pid,sop,ah;**

**}process;**

**typedef struct**

**{**

**int hid,soh,ex;**

**}hole;**

**int i,j,k;**

**void Table(process p[],hole h[],int np);**

**void main()**

**{**

**int min=9999,np,nh,max=-1;**

**printf ("Enter the number of Holes: ",&nh);**

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

**hole h[nh];**

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

**{**

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

**scanf ("%d",&h[i].soh);**

**h[i].hid=i+1;**

**h[i].ex=1;**

**}**

**printf ("\nEnter the number of Process: ",&np);**

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

**process p[np];**

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

**{**

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

**scanf ("%d",&p[i].sop);**

**p[i].pid=i+1;**

**p[i].ah=-1;**

**}**

**printf("\n\n Worst Fit Mamory Allocation");**

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

**{**

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

**{**

**if (h[k].soh>max && h[k].ex==1)**

**{**

**max=h[k].soh;**

**j=k;**

**}**

**}**

**if (max>=p[i].sop)**

**{**

**p[i].ah=j;**

**h[p[i].ah].ex=0;**

**}**

**max=0;**

**}**

**Table(p,h,np);**

**}**

**void Table(process p[],hole h[],int np)**

**{**

**int sum=0;**

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

**for(i=0; i<75; i++)**

**printf("-");**

**printf("\n Process | Process | Allocated | Hole | Left Over |");**

**printf("\n | Actual Size | Hole | Actual Size | Space |\n");**

**for(i=0; i<75; i++)**

**printf("-");**

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

**{**

**if (p[i].ah!=-1)**

**{**

**printf("\n P%d | %-4d | H%d | %-4d | %-4d |",p[i].pid,p[i].sop,p[i].ah+1,h[p[i].ah].soh,h[p[i].ah].soh-p[i].sop);**

**sum+=h[p[i].ah].soh-p[i].sop;**

**}**

**else**

**{**

**printf("\n P%d | %-4d | NOT ALLOCATED | - | - |",p[i].pid,p[i].sop);**

**}**

**}**

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

**for(i=0; i<75; i++)**

**printf("-");**

**printf("\n\nAccording to the Worst Fit Memory Allocation");**

**printf("\nTotal left over space : %d\n\n",sum);**

**}**

**Output:**

