```
BEST FIT ,WORST FIT ,FIRST FIT:
                                                      for(j=0;j<nm;j++)
#include<stdio.h>
#include<conio.h>
                                                      if(flag[j]==0 && msize[j]>=psize[i])
void accept(int a[],int n)
                                                      flag[j]=1;
int i;
                                                      itot=itot+msize[j]-psize[i];
for(i=0;i< n;i++)
                                                      break;
scanf("%d",&a[i]);
                                                      }
                                                      if(j==nm)
                                                      printf("\n\nTHERE IS NO SPACE FOR
void display(int a[],int n)
                                                      PROCESS %d ",i);
int i;
                                                      for(i=0;i<nm;i++)
printf("\n\n");
for(i=0;i< n;i++)
                                                      if(flag[i]==0)
                                                      etot=etot+msize[i];
printf("\t%d ",a[i]);
                                                      printf("\n\nPROCESSES::");
                                                      display(psize,np);
                                                      printf("\n\nMEMORY HOLES::");
void sort(int a[],int n)
                                                      display(msize,nm);
int i,j,temp;
                                                      printf("\n\nTOTAL SUM OF INTERNAL
                                                      FRAGMENTATION = %d ",itot);
for(i=0;i< n-1;i++)
                                                      printf("\n\nTOTAL SUM OF EXTERNAL
for(j=0;j< n-1;j++)
                                                      FRAGMENTATION = %d ",etot);
if(a[j]>a[j+1])
                                                      void best_fit(int psize[],int np,int msize[],int
                                                      nm)
temp=a[j];
                                                      {
a[j]=a[j+1];
                                                      int i,j,itot,etot,temp[30],flag[30]={0};
a[j+1]=temp;
                                                      itot=etot=0;
                                                      for(i=0;i<nm;i++)
                                                      temp[i]=msize[i];
}
                                                      sort(temp,nm);
                                                      for(i=0;i<np;i++)
void first_fit(int psize[],int np,int msize[],int
nm)
                                                      for(j=0;j<nm;j++)
nt i,j,itot,etot,flag[30]={0};
                                                      if(flag[j]==0 && temp[j]>=psize[i])
tot=etot=0;
for(i=0;i<np;i++)
                                                      flag[j]=1;
                                                      itot=itot+temp[j]-psize[i];
```

```
break;
}
                                                  for(i=0;i<nm;i++)
if(j==nm)
                                                  if(flag[i]==0)
printf("\n\nTHERE IS NO SPACE FOR
                                                  etot=etot+temp[i];
PROCESS %d ",i);
                                                  printf("\n\nPROCESSES::");
for(i=0;i<nm;i++)
                                                  display(psize,np);
                                                  printf("\n\nMEMORY HOLES::");
if(flag[i]==0)
                                                  display(temp,nm);
etot=etot+temp[i];
                                                  printf("\n\nTOTAL SUM OF INTERNAL
                                                  FRAGMENTATION = %d ",itot);
printf("\n\nPROCESSES::");
                                                  printf("\n\nTOTAL SUM OF EXTERNAL
display(psize,np);
                                                  FRAGMENTATION = %d ",etot);
printf("\n\nMEMORY HOLES::");
display(temp,nm);
                                                  void main()
printf("\n\nTOTAL SUM OF INTERNAL
FRAGMENTATION = %d ",itot);
                                                  int ch,np,nm,psize[30],msize[30];
printf("\n\nTOTAL SUM OF EXTERNAL
                                                  clrscr();
FRAGMENTATION = %d ",etot);
                                                  printf("\nENTER NO OF PROCESSES::");
                                                  scanf("%d",&np);
void worst_fit(int psize[],int np,int msize[],int
                                                  printf("\n\nENTER SIZES OF
nm)
                                                  PROCESSES::");
{
                                                  accept(psize,np);
int i,j,itot,etot,temp[30],flag[30]={0};
                                                  printf("\nENTER NO MEMORY HOLES::");
itot=etot=0;
                                                  scanf("%d",&nm);
                                                  printf("\n\nENTER SIZES OF MEMORY
for(i=0;i<nm;i++)
temp[i]=msize[i];
                                                  HOLES::");
sort(temp,nm);
                                                  accept(msize,nm);
for(i=0;i<np;i++)
                                                  while(1)
for(j=nm-1;j>=0;j-)
                                                  printf("\n\n\t\t**MAIN MENU**");
                                                  printf("\n\n\tMEMORY MANAGEMENT");
if(flag[j]==0 && temp[j]>=psize[i])
                                                  printf("\n\n\t1.FIRST FIT");
{
                                                  rintf("\n\n\t2.BEST FIT");
flag[j]=1;
                                                  rintf("\n\n\t3.WORST FIT");
itot=itot+temp[j]-psize[i];
                                                  rintf("\n\n\t4.QUIT");
break;
                                                  printf("\n\nENTER YOUR CHOICE::");
}
                                                  scanf("%d",&ch);
                                                  switch(ch)
if(j==nm)
printf("\n\nTHERE IS NO SPACE FOR
                                                  case 1:
PROCESS %d ",i);
                                                  printf("\n\nFIRST FIT::\n");
```

first_fit(psize,np,msize,nm); break;	5 15 10 35 25 20 25
<pre>case 2: printf("\n\n\tBEST FIT::\n"); best_fit(psize,np,msize,nm); break; case 3: printf("\n\n\tWORST FIT::\n");</pre>	TOTAL SUM OF INTERNAL FRAGMENTATION = 30 TOTAL SUM OF EXTERNAL FRAGMENTATION = 60
worst_fit(psize,np,msize,nm); break;	ENTER YOUR CHOICE::2
case 4: exit(0); default:	BEST FIT:: THERE IS NO SPACE FOR PROCESS 4 PROCESSES::
printf("\n\nPLEASE ENTER CORRECT CHOICE!!");	10 20 15 30 45 MEMORY HOLES::
<pre>} getch(); } </pre>	5 10 15 20 25 25 35 TOTAL SUM OF INTERNAL FRAGMENTATION = 5
OUTPUT: ENTER NO OF PROCESSES::5 ENTER SIZES OF PROCESSES::10 20 15	TOTAL SUM OF EXTERNAL FRAGMENTATION = 55 ENTER YOUR CHOICE::3
30 45 ENTER NO MEMORY HOLES::7	WORST FIT:: PROCESSES::
ENTER SIZES OF MEMORY HOLES::5 15 10 35 25 20 25	10 20 15 30 45 MEMORY HOLES::
MAIN MENU	5 10 15 20 25 25 35
MEMORY MANAGEMENT 1.FIRST FIT 2.BEST FIT 3.WORST FIT 4.QUIT	TOTAL SUM OF INTERNAL FRAGMENTATION = 40 TOTAL SUM OF EXTERNAL FRAGMENTATION = 50
ENTER YOUR CHOICE::1 FIRST FIT:: THERE IS NO SPACE FOR PROCESS 3 THERE IS NO SPACE FOR PROCESS 4 PROCESSES:: 10 20 15 30 45 MEMORY HOLES::	