Fitting algorithms:

First fit:

#include<stdio.h>

main()

{

int ps,bs,blocks[20],process[20],allocate[10],p[10],i,j;

printf("enter no of processors");

scanf("%d",&ps);

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

{

printf("process%d:",i+1);

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

p[i]=i+1;

}

printf("enter no of blocks");

scanf("%d",&bs);

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

{

printf("blocks%d:",i+1);

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

}

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

{

allocate[i]=-1;

}

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

{

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

{

if(blocks[j]>=process[i])

{

allocate[i]=j;

blocks[j]=0;

break;

}

}

}

printf("process\tprocess\tblocks\n");

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

{

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

if(allocate[i]!=-1)

{

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

}

else

{

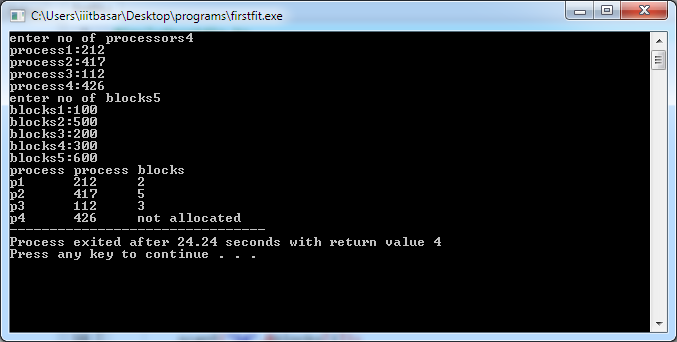
printf("not allocated");

}

}

}

o/p:



Bestfit:

#include<stdio.h>

main()

{

int i,j,ps,bs,process[10],blocks[10],allocate[10],p[i];

printf("enter no of process size");

scanf("%d",&ps);

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

{

printf("process%d:",i+1);

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

p[i]=i+1;

}

printf("enter no of blocksize");

scanf("%d",&bs);

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

{

printf("blocks%d:",i+1);

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

}

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

{

allocate[i]=-1;

}

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

{

int min,i1,j1,k;

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

{

if(blocks[j]>=process[i])

{

int i1=process[i];

int j1=blocks[j];

for(k=0;k<bs;k++)

{

if(blocks[k]>i1&&blocks[k]<=j1)

{

min=k;

}

}

allocate[i]=min;

blocks[min]=0;

break;

}

}

}

printf("process\tprocess\tblocks\n");

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

{

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

if(allocate[i]!=-1)

{

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

}

else

{

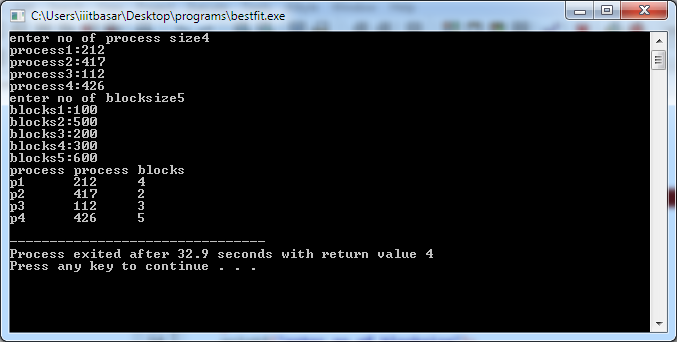
printf("not allocated");

}

}

}

o/p:



Worst fit:

#include<stdio.h>

main()

{

int i,j,ps,bs,process[10],blocks[10],allocate[10],p[i];

printf("enter no of process size");

scanf("%d",&ps);

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

{

printf("process%d:",i+1);

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

p[i]=i+1;

}

printf("enter no of blocksize");

scanf("%d",&bs);

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

{

printf("blocks%d:",i+1);

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

}

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

{

allocate[i]=-1;

}

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

{

int max,i1,j1,k;

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

{

if(blocks[j]>=process[i])

{

int i1=process[i];

int j1=blocks[j];

for(k=0;k<bs;k++)

{

if(blocks[k]>i1&&blocks[k]>=j1)

{

max=k;

}

}

allocate[i]=max;

blocks[max]=0;

break;

}

}

}

printf("process\tprocess\tblocks\n");

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

{

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

if(allocate[i]!=-1)

{

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

}

else

{

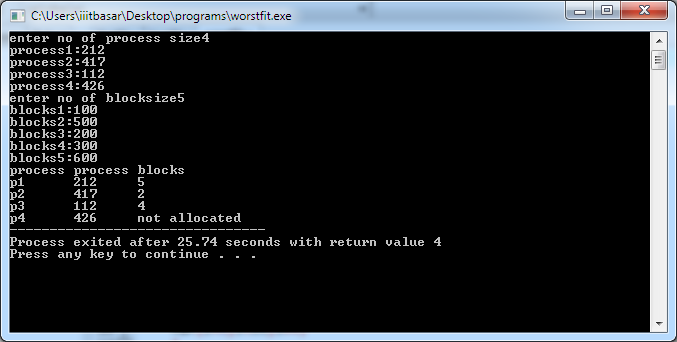
printf("not allocated");

}

}

}

o/p:

****

**Fifo:**

**#include<stdio.h>**

**main()**

**{**

**int n,no,j,i,count=0,k,a[30],avail,frame[30];**

**printf("enter no of pages\n");**

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

**printf("enter page numbers:");**

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

**{**

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

**}**

**printf("enter no of frames");**

**scanf("%d",&no);**

**for(i=0;i<no;i++)**

**{**

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

**j=0;**

**}**

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

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

**{**

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

**avail=0;**

**for(k=0;k<no;k++)**

**{**

**if(frame[k]==a[i])**

**{**

**avail=1;**

**}**

**}**

**if(avail==0)**

**{**

**frame[j]=a[i];**

**count++;**

**j=(j+1)%no;**

**for(k=0;k<no;k++)**

**{**

**printf("%d",frame[k]);**

**}**

**}**

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

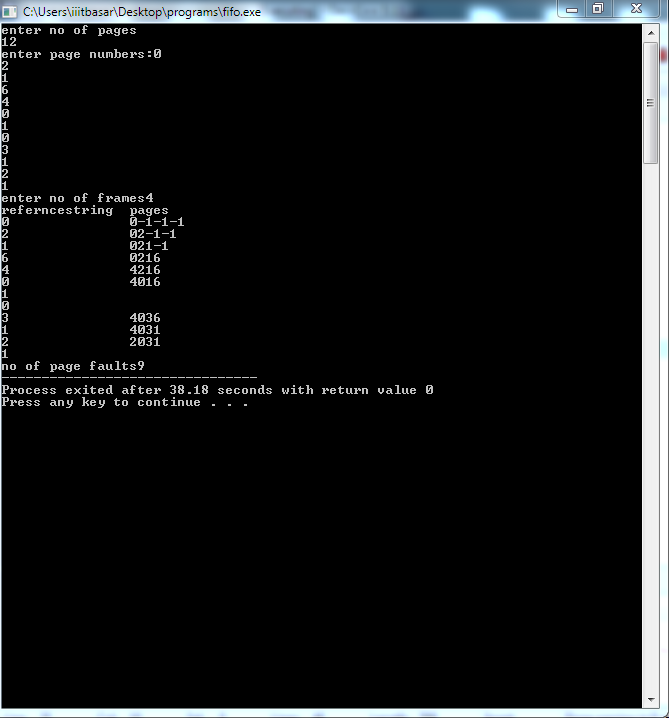
**}**

**printf("no of page faults%d",count);**

**return 0;**

**}**

**o/p:**

****