13. Implement the following memory allocation methods for fixed partition

a) First fit b) Worst fit c) Best fit

**AIM:** To write a C program for implementation memory allocation methods for fixed partition using first fit.

ALGORITHM: Step 1:Define the max as 25.

Step 2: Declare the variable frag[max],b[max],f[max],i,j,nb,nf,temp, highest=0, bf[max],ff[max].

Step 3: Get the number of blocks,files,size of the blocks using for loop.

Step 4: In for loop check bf[j]!=1, if so temp=b[j]-f[i]

Step 5: Check highest

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

#define max 25

void main()

{

int frag[max],b[max],f[max],i,j,nb,nf,temp,highest=0;

static int bf[max],ff[max];

printf("\n\tMemory Management Scheme - Worst Fit");

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

scanf("%d",&nb);

printf("Enter the number of files:");

scanf("%d",&nf);

printf("\nEnter the size of the blocks:-\n");

for(i=1;i<=nb;i++)

{

printf("Block %d:",i);

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

}

printf("Enter the size of the files :-\n");

for(i=1;i<=nf;i++)

{

printf("File %d:",i);

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

}

for(i=1;i<=nf;i++)

{

for(j=1;j<=nb;j++)

{

if(bf[j]!=1) //if bf[j] is not allocated

{

temp=b[j]-f[i];

if(temp>=0)

if(highest<temp)

{

ff[i]=j;

highest=temp;

}

}

}

frag[i]=highest;

bf[ff[i]]=1;

highest=0;

}

printf("\nFile\_no:\tFile\_size :\tBlock\_no:\tBlock\_size:\tFragement");

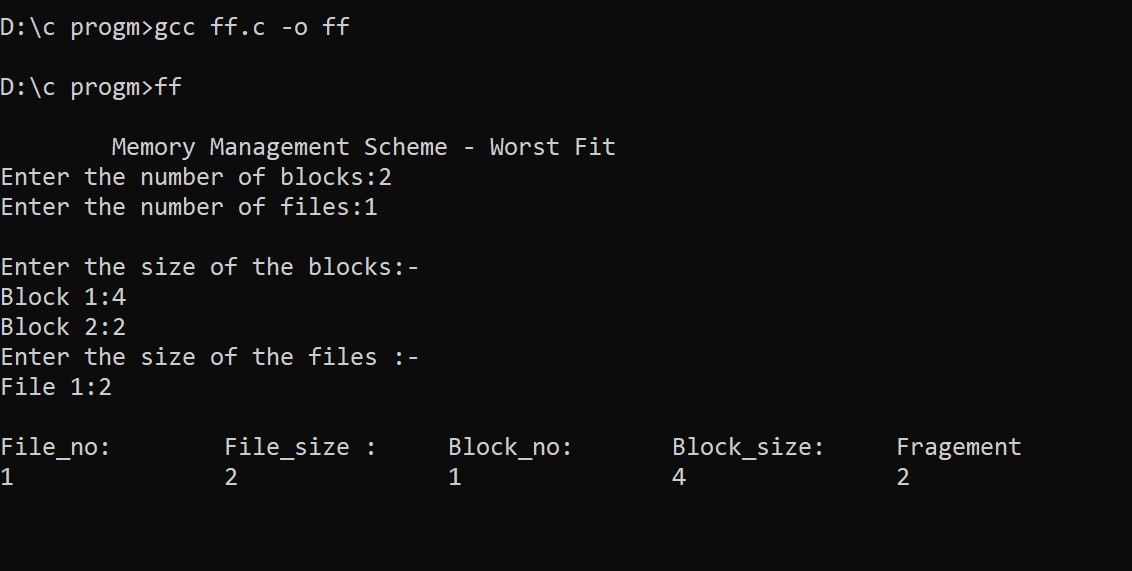
for(i=1;i<=nf;i++)

printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);

getch();

}

**Out put:**

****

**AIM**: To write a C program for implementation of FCFS and SJF scheduling algorithms.

**ALGORITHM:**

Step 1:Define the max as 25.

Step 2: Declare the variable frag[max],b[max],f[max],i,j,nb,nf,temp, highest=0, bf[max],ff[max].

Step 3: Get the number of blocks,files,size of the blocks using for loop.

Step 4: In for loop check bf[j]!=1, if so temp=b[j]-f[i]

Step 5: Check temp>=0,if so assign ff[i]=j break the for loop.

Step 6: Assign frag[i]=temp,bf[ff[i]]=1;

Step 7: Repeat step 4 to step 6.

Step 8: Print file no,size,block no,size and fragment.

Step 9: Stop the program.

**PROGRAM:**

**#include<stdio.h>**

**#include<conio.h>**

**#define max 25**

**void main()**

**{**

**int frag[max],b[max],f[max],i,j,nb,nf,temp;**

**static int bf[max],ff[max];**

**printf("\n\tMemory Management Scheme - First Fit");**

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

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

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

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

**printf("\nEnter the size of the blocks:-\n");**

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

**{**

**printf("Block %d:",i);**

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

**}**

**printf("Enter the size of the files :-\n");**

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

**{**

**printf("File %d:",i);**

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

**}**

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

**{**

**for(j=1;j<=nb;j++)**

**{**

**if(bf[j]!=1)**

**{**

**temp=b[j]-f[i];**

**if(temp>=0)**

**{**

**ff[i]=j;**

**break;**

**}**

**}**

**}**

**frag[i]=temp;**

**bf[ff[i]]=1;**

**}**

**printf("\nFile\_no:\tFile\_size :\tBlock\_no:\tBlock\_size:\tFragement");**

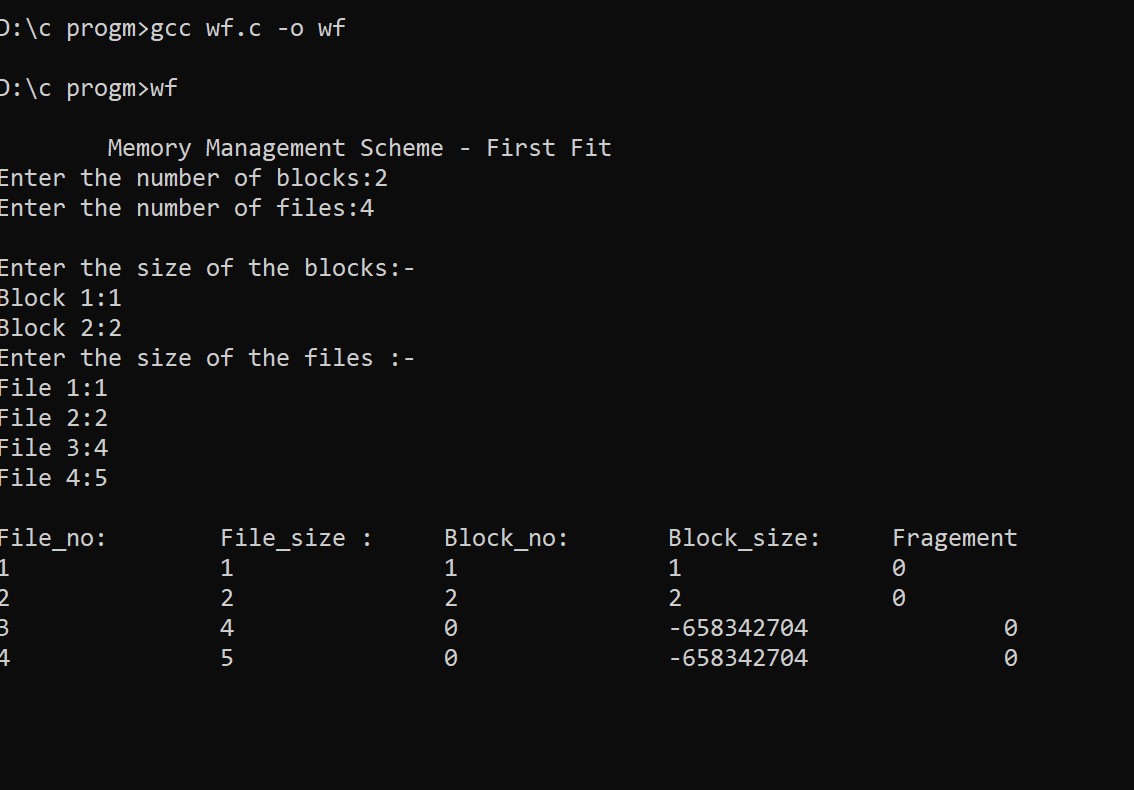
**for(i=1;i<=nf;i++)**

**printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);**

**getch();**

**}**

**Output:**

****

**AIM:** To write a C program for implementation of FCFS and SJF scheduling algorithms.

**ALGORITHM:**

Step 1:Define the max as 25.

Step 2: Declare the variable frag[max],b[max],f[max],i,j,nb,nf,temp, highest=0, bf[max],ff[max].

Step 3: Get the number of blocks,files,size of the blocks using for loop.

Step 4: In for loop check bf[j]!=1, if so temp=b[j]-f[i]

Step 5: Check lowest>temp,if so assign ff[i]=j,highest=temp

Step 6: Assign frag[i]=lowest, bf[ff[i]]=1,lowest=10000

Step 7: Repeat step 4 to step 6.

Step 8: Print file no,size,block no,size and fragment.

Step 9: Stop the program.

**PROGRAM:**

**#include<stdio.h>**

**#include<conio.h>**

**#define max 25**

**void main()**

**{**

**int frag[max],b[max],f[max],i,j,nb,nf,temp,lowest=10000;**

**static int bf[max],ff[max];**

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

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

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

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

**printf("\nEnter the size of the blocks:-\n");**

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

**{**

**printf("Block %d:",i);**

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

**}**

**printf("Enter the size of the files :-\n");**

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

**{**

**printf("File %d:",i);**

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

**}**

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

**{**

**for(j=1;j<=nb;j++)**

**{**

**if(bf[j]!=1)**

**{**

**temp=b[j]-f[i];**

**if(temp>=0)**

**if(lowest>temp)**

**{**

**ff[i]=j;**

**lowest=temp;**

**}**

**}**

**}**

**frag[i]=lowest;**

**bf[ff[i]]=1;**

**lowest=10000;**

**}**

**printf("\nFile No\tFile Size \tBlock No\tBlock Size\tFragment");**

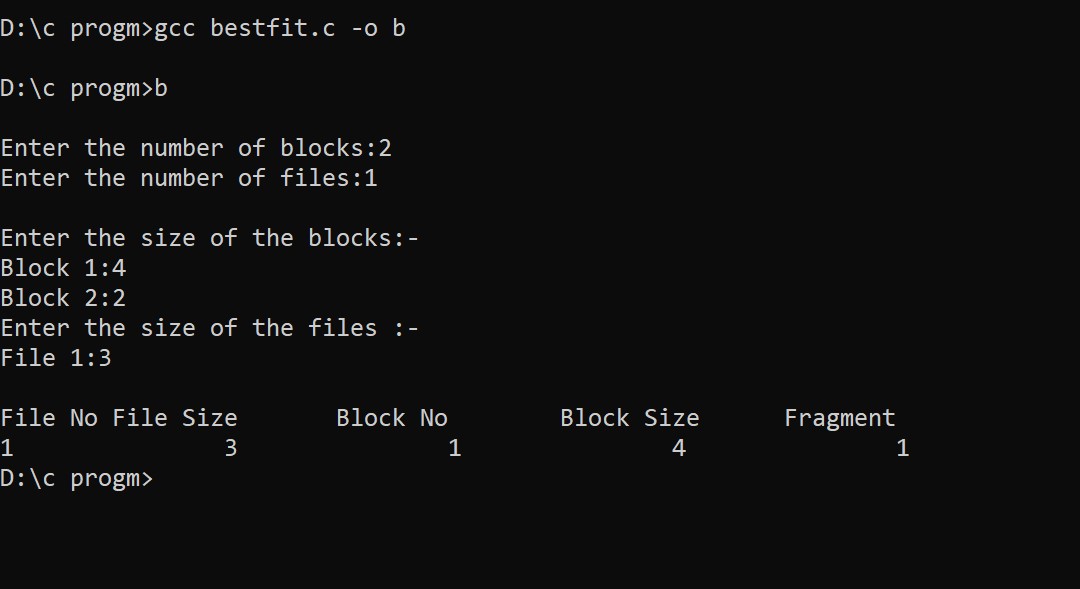
**for(i=1;i<=nf && ff[i]!=0;i++)**

**printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);**

**getch();**

**}**

**Output:**

****