S.No: 8	Exp. Name: *Write a program to Implementation of Contiguous allocation technique: - Best-Fit*	Date:

Aim:

Write a program to Implementation of Contiguous allocation technique: - Best-Fit

Source Code:

bestFit.c

```c
#include<stdio.h>
#include<conio.h>
#define max 25
int main()
{
    int frag[max],b[max],f[max],i,j,nb,nf,temp,lowest=10000;
    static int bf[max],ff[max];
    printf("Memory Management Scheme for contigus memeory allocation - Best Fit\n");
    printf("Enter the number of blocks:");
    scanf("%d",&nb);
    printf("Enter the number of files:");
    scanf("%d",&nf);
    printf("Enter 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("File No\tFile Size \tBlock No\tBlock Size\tFragment");
    for(i=1;i<=nf && ff[i]!=0;i++)
    printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);
    return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1						
User Output						
Memory Management Scheme for contigus memeory allocation - Best Fit 3						
Enter the number of blocks: 3						
Enter the number of files: 2						
Enter the size of the blocks:- 5						
Block 1: 5						
Block 2: 1						
Block 3: 4						
Enter the size of the files :- 3						
File 1: 3						
File 2: 4						
File No File Size	Block No	Block Size	Fragment1	3	3	4	12	4