

Write a C program to simulate the following contiguous memory allocation techniques:

a) Worst-fit

b) Best-fit

c) First-fit

:-1BM21CS232

```
#include<stdio.h>
#define max 25
int
frag[max],b[max],f[max],i,j,nb,nf,temp,highest=0,lowest=10000,ch;
static int bf[max],ff[max];
void firstfit();
void bestfit();
void worstfit();
void main()
{
    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]);
    }
    printf("Enter the choice: 1.First fit 2.Best Fit 3.Worst fit\n");
    scanf("%d",&ch);
    switch(ch)
    {
        case 1: firstfit();
            break;
        case 2: bestfit();
            break;
        case 3: worstfit();
            break;
        default: printf("Invalid choice");
    }
}

void firstfit()
{
    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]);
}

```

```

void bestfit()
{
    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]);
}

```

```

void worstfit()
{
    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(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]);
}

```

OutPut:

The screenshot displays a C++ IDE with the following components:

- Source Code (Mem-Management.c):**

```

1 #include<stdio.h>
2 #define max 25
3 int frag[max],b[max],f[max],i,j,nb,ff[max];
4 static int bf[max],ff[max];
5 void firstfit();
6 void bestfit();
7 void worstfit();
8 void main()
9 {
10     printf("\nEnter the number of blocks:");
11     scanf("%d",&nb);
12     printf("Enter the number of files:");
13     scanf("%d",&nf);
14     printf("Enter the size of the files:");
15     for(i=1;i<=nf;i++)
16     {
17         printf("Block %d:",i);
18         scanf("%d",&b[i]);
19     }
20     printf("Enter the size of the files:");
21     for(i=1;i<=nf;i++)
22     {
23         printf("File %d:",i);
24         scanf("%d",&f[i]);
25     }

```
- Execution Output:**

```

Enter the number of files:3
Enter the size of the blocks:-
Block 1:10
Block 2:4
Block 3:20
Block 4:18
Block 5:7
Block 6:9
Block 7:12
Block 8:15
Enter the size of the files :-
File 1:12
File 2:10
File 3:9
Enter the choice: 1.First fit 2.Best Fit 3.Worst fit
1
File_no:      File_size :      Block_no:      Block_size:      Fragement
1             12             3             20             8
2             10             1             10             0
3             9              4             18             9
-----
Process exited after 80.75 seconds with return value 3
Press any key to continue . . .

```
- Compiler Output:**

```

Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Admin\Desktop\18M21CS232\Mem-Management.exe
- Output Size: 131.5517578125 KiB
- Compilation Time: 0.14s

```