Week 3

Program

write a c program to simulate the following contigous memory allocation techniques.

(a) worstfit (b) best fit (c) first fit.

```
#include <stdio.h>
#define MAX 25.
void firstfit(int nb, int nf, int b[], int f[]){
    int frag[MAX], bf[MAX] = {0}, ff[MAX] = {0};
    int i, j, temp;
    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;
                    frag[i] = temp;
                    bf[j] = 1;
                    break;
                }
            }
        }
    }
printf("\n In Memory Management Scheme - First fit \n");
printf("File_no \t File Size : \t Block_no \t Fragment \n");
for(i=1; i<=nf; i++){
    printf("%d \t\t %d \t\t", i, f[i]);
    if(ff[i] != 0){
        printf("%d \t\t %d \t\t %d \n",
            ff[i], b[ff[i]], frag[i]);
    }
}
```

```
        else
            printf("Not Allocated \n");
    }
}
void bestfit(int nb, int nf, int b[], int f[]){
    int frag[MAX], bf[MAX]={0}, ff[MAX]={0};
    int i, j, temp, lowest = 10000;
    for(i=1; i<=nf; i++){
        for(j=1; j<=nb; j++){
            if(bf[j] != 1){
                temp = b[j] - f[i];
                if(temp >= 0 && lowest > temp){
                    ff[i] = j;
                    lowest = temp;
                }
            }
        }
        frag[i] = lowest;
        bf[ff[i]] = 1;
        lowest = 10000;
    }
    printf("\nMemory Management Scheme - Best Fit\n");
    printf("File No \t File Size \t Block No \t Block Size
        \t Fragment \n");
    for(i=1; i<=nf; i++){
        printf("%d \t\t %d \t\t %d \t\t", i, f[i]);
        if(ff[i] != 0){
            printf("%d \t\t %d \t\t %d \n",
                ff[i], b[ff[i]], frag[i]);
        }else{
            printf("Not Allocated\n");
```

```
}
}
}

void worstfit(int nb, int nf, int b[], int f[]){
    int frag[max], bf[max]={0}, ff[max]={0};
    int i, j, temp, highest=0;

    for(i=1; i<=nf; i++){
        for(j=1; j<=nb; j++){
            if(bf[j]!=1){
                temp = b[j]-f[i];
                if(temp>=0 && highest<temp){
                    ff[i]=j;
                    highest=temp;
                }
            }
        }
        frag[i]=highest;
        bf[ff[i]]=1;
        highest=0;
    }

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

printf("\n File_no: \t File_size \t Block_no: \t Block_size \t Fragment \n");

for(i=1; i<=nf; i++){
    printf("%d \t\t %d \t\t %d \t\t %d", i, f[i]);
    if(ff[i]!=0){
        printf("%d \t\t %d \t\t %d \t\t ",
            ff[i], b[ff[i]], frag[i]);
```

```
    } else {
        printf("Not Allocated \n");
    }
}

int main() {
    int b[MAX], f[MAX], nb, nf;
    printf("\n Enter no. of blocks:\n");
    scanf("%d", &nb);
    printf("\n Enter no. of files:\n");
    scanf("%d", &nf);
    printf("\n Enter size of blocks:\n");
    for(int i=1; i<=nb; i++) {
        printf("Block %d :", i);
        scanf("%d", &b[i]);
    }
    printf("Enter the size of files:\n");
    for(int i=1; i<=nf; i++) {
        printf("File %d :", i);
        scanf("%d", &f[i]);
    }
    int b1[MAX], b2[MAX], b3[MAX];
    for(int i=1; i<=nb; i++) {
        b1[i] = b[i];
        b2[i] = b[i];
        b3[i] = b[i];
    }
    firstFit(nb, nf, b1, f);
    bestFit(nb, nf, b2, f);
    WoorstFit(nb, nf, b3, f);
}
```

Output:

Enter No. of Blocks : 6

Enter No. of file : 4

Enter Size of Blocks :

Block 1 : 200

Block 2 : 400

Block 3 : 600

Block 4 : 500

Block 5 : 300

Block 6 : 250

Enter the size of file

File 1 : 357

File 2 : 210

File 3 : 468

File 4 : 491

Sam 4/7/24

Memory management scheme - First Fit.

File No:	File-Size	Block no.	Block Size	Fragment
1	357	2	400	43
2	210	5	600	390
3	468	4	500	32
4	491	Not Allocated		

Memory management scheme - Best Fit.

File No.	File Size	Block No	Block Size	Fragment
1	357	2	400	43
2	210	6	250	40
3	468	4	500	32
4	491	3	600	109

Memory management scheme - Worst Fit.

File No	File Size	Block No.	Block Size	Fragment
1	357	5	600	243
2	210	4	500	290
3	468	Not Allocated		
4	491	Not Allocated		