

Write a C program to simulate the following contiguous memory allocation techniques  
 a) Worst-fit                      b) Best-fit                      c) First-fit

```
→ #include <stdio.h>
#include <stdlib.h>
#define MAX 25
```

```
void firstfit (int nb, int nf, int b[], int f[]) {
    int ff[MAX] = {0};
    int allocated[MAX] = {0};
    for (int i = 0; i < nf; i++) {
        ff[i] = -1;
        for (int j = 0; j < nb; j++) {
            if (allocated[j] == 0 && b[j] >= f[i]) {
                ff[i] = j;
                allocated[j] = 1;
                break;
            }
        }
    }
}
```

```
printf("\n File no: \t File size: \t Block no: \t\n\n");
```

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

```
else
    printf("\n %d \t %d \t - \t -",
        i+1, f[i]);
}
```

```

void bestfit(int nb, int nf, int b[], int f[]){
    int ff[MAX] = {0};
    int allocated[MAX] = {0};
    for(int i=0; i<nf; i++){
        int best = -1;
        ff[i] = -1;
        for(int j=0; j<nb; j++){
            if(allocated[j]==0 && b[j]>=f[i]){
                if(best == -1 || b[j]<b[best])
                    best = j;
            }
        }
        if(best != -1){
            ff[i] = best;
            allocated[best] = 1;
        }
    }
    printf("\n File no: | File size: | Block no: | Block size: ");
    for(int i=0; i<nf; i++){
        if(ff[i] != -1)
            printf("\n %d | %d | %d | %d", i+1, f[i], ff[i]+1, b[ff[i]]);
        else
            printf("\n %d | %d | - | -", i+1, f[i]);
    }
}

```

```

void worstfit(int nb, int nf, int b[], int f[]){
    int ff[MAX] = {0};
    int allocated[MAX] = {0};

```

```

for (int i=0 ; i < nf ; i++) {
    int worst = -1;
    if (f[i] == -1;
    for (int j=0 ; j < nb ; j++) {
        if (allocated[j] == 0 && b[j] >= f[i]) {
            if (worst == -1 || b[j] > b[worst])
                worst = j;
        }
    }
    if (worst != -1) {
        f[i] = worst;
        allocated[worst] = 1;
    }
}

```

```

printf("In File no: \t File size: \t Block no: \t\n\n");

```

```

for (int i=0 ; i < nf ; i++) {
    if (f[i] != -1)
        printf("In %d \t \t %d \t \t %d \t \t %d\n",
            i+1, f[i], f[i]+1, b[f[i]]);
    else
        printf("In %d \t \t %d \t \t - \t \t -\n",
            i+1, f[i]);
}
}

```

```

int main() {

```

```

    int nb, nf, choice;

```

```

    printf("Memory management Scheme");

```

```

    printf("\n Enter no of Blocks");

```

```

    scanf("%d", &nb);

```

```

    printf("Enter no of files");

```

```

    scanf("%d", &nf);

```



```

int b[nb], f[nf];
printf("\n Enter size of blocks:\n");
for(int i=0; i<nb; i++){
    printf("Block i.d :", i+1);
    scanf("%d", &b[i]);
}

printf("\n Enter the size of files:\n");
for(int i=0; i<nf; i++){
    printf("File i.d :", i+1);
    scanf("%d", &f[i]);
}

while(1){
    printf("\n 1: first fit  2: Best fit  3: Worst fit\n");
    printf("4: Exit\n");
    printf("Enter your choice:");
    scanf("%d", &choice);
    switch(choice){
        case 1: printf("\n It Memory management\n");
                 printf("Scheme - First Fit\n");
                 firstfit(nb, nf, b, f);
                 break;
        case 2: printf("\n MMS - Best Fit\n");
                 Bestfit(nb, nf, b, f);
                 break;
        case 3: printf("\n MMS - Worst Fit\n");
                 worstfit(nb, nf, b, f);
                 break;
        case 4: exit(0);
                 break;
        default: printf("Invalid choice\n");
                 break;
    }
}

```

→ Output:-

Memory Management Scheme  
Enter no. of blocks: 6  
Enter no. of files: 4

Enter size of blocks:

Block 1 : 200

Block 2 : 400

Block 3 : 600

Block 4 : 500

Block 5 : 300

Block 6 : 250

Enter size of files:

File 1: 357

File 2: 210

File 3: 468

File 4: 491

1: First Fit

2: Best Fit

3: Worst Fit

4: Exit

Enter your choice: 1

Memory management scheme - First Fit

File no	File size	Block no	Block size
1	357	2	400
2	210	3	600
3	468	4	500
4	491	-	-

Enter your choice : 2

Memory management Scheme : Best Fit

File no	File size	Block no	Block size
1	357	2	400
2	210	6	250
3	468	4 . .	500
4	491	3 . .	600

Enter your choice : 3

Memory management Scheme : Worst Fit

File no	File size	Block no	Block size
1	357	3	600
2	210	4	500
3	468	-	-
4	491	-	-

Sum  
417124