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

Code:

```
#include <stdio.h>

#define MAX 10

int main() {
    int blockSize[MAX], fileSize[MAX];
    int blockCount, fileCount;
    int choice;

    printf("Memory Management Scheme:\n");
    printf("Enter the number of blocks: ");
    scanf("%d", &blockCount);
    printf("Enter the number of files: ");
    scanf("%d", &fileCount);

    printf("Enter the size of the blocks:\n");
    for (int i = 0; i < blockCount; i++) {
        printf("Block %d: ", i + 1);
        scanf("%d", &blockSize[i]);
    }

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

    do {
        printf("\n1. First Fit\n2. Best Fit\n3. Worst Fit\n4. Exit\nEnter your choice: ");
        scanf("%d", &choice);

        int allocation[MAX];
        int tempBlockSize[MAX];
        int allocated[MAX]={0};

        for (int i = 0; i < blockCount; i++)
            tempBlockSize[i] = blockSize[i];
```

```
for (int i = 0; i < fileCount; i++)
    allocation[i] = -1;

switch (choice) {
    case 1:
        printf("\n\tMemory Management Scheme – First Fit\n");
        for (int i = 0; i < fileCount; i++) {
            for (int j = 0; j < blockCount; j++) {
                if (!allocated[j] && tempBlockSize[j] >= fileSize[i]) {
                    allocation[i] = j;
                    allocated[j]=1;
                    break;
                }
            }
        }
        break;

    case 2:
        printf("\n\tMemory Management Scheme – Best Fit\n");
        for (int i = 0; i < fileCount; i++) {
            int bestIdx = -1;
            for (int j = 0; j < blockCount; j++) {
                if (!allocated[j] && tempBlockSize[j] >= fileSize[i]) {
                    if (bestIdx == -1 || tempBlockSize[j] < tempBlockSize[bestIdx]) {
                        bestIdx = j;
                    }
                }
            }
            if (bestIdx != -1) {
                allocation[i] = bestIdx;
                allocated[bestIdx]=1;
            }
        }
        break;

    case 3:
        printf("\n\tMemory Management Scheme – Worst Fit\n");
        for (int i = 0; i < fileCount; i++) {
            int worstIdx = -1;
            for (int j = 0; j < blockCount; j++) {
                if (!allocated[j] && tempBlockSize[j] >= fileSize[i]) {
                    if (worstIdx == -1 || tempBlockSize[j] > tempBlockSize[worstIdx]) {
                        worstIdx = j;
```

```
                    }
                }
            }
            if (worstIdx != -1) {
                allocation[i] = worstIdx;
                allocated[worstIdx]=1;
            }
        }
        break;

    case 4:
        return 0;

    default:
        printf("Invalid choice\n");
    }

    // Print allocation
    if (choice >= 1 && choice <= 3) {
        printf("File_no:\tFile_size :\tBlock_no:\tBlock_size:\n");
        for (int i = 0; i < fileCount; i++) {
            printf("%d\t\t%d\t\t", i + 1, fileSize[i]);
            if (allocation[i] != -1)
                printf("%d\t\t%d\n", allocation[i] + 1, blockSize[allocation[i]]);
            else
                printf("Not Allocated\t-\n");
        }
    }

} while (choice != 4);

return 0;
}
```

Output:

```
Memory Management Scheme:
Enter the number of blocks: 5
Enter the number of files: 4
Enter the size of the blocks:
Block 1: 100
Block 2: 500
Block 3: 200
Block 4: 300
Block 5: 600
Enter the size of the files:
File 1: 212
File 2: 417
File 3: 112
File 4: 426

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               212             2               500
2               417             5               600
3               112             3               200
4               426             Not Allocated   -
```

```
1. First Fit
2. Best Fit
3. Worst Fit
4. Exit
Enter your choice: 2

        Memory Management Scheme - Best Fit
File_no:        File_size :     Block_no:       Block_size:
1               212             4               300
2               417             2               500
3               112             3               200
4               426             5               600

1. First Fit
2. Best Fit
3. Worst Fit
4. Exit
Enter your choice: 3

        Memory Management Scheme - Worst Fit
File_no:        File_size :     Block_no:       Block_size:
1               212             5               600
2               417             2               500
3               112             4               300
4               426             Not Allocated   -

1. First Fit
2. Best Fit
3. Worst Fit
4. Exit
Enter your choice:
```