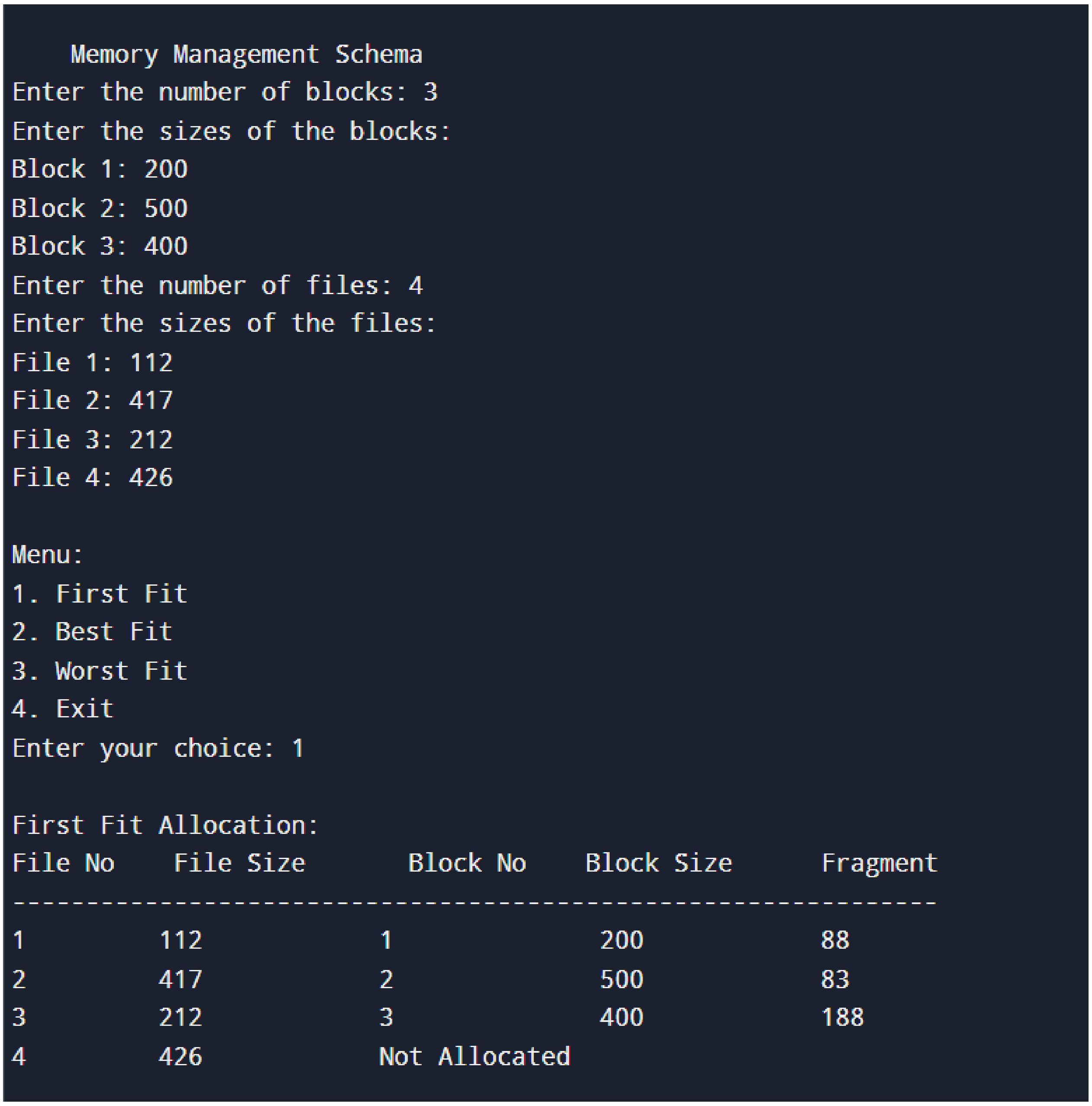
Code :

```
#include<stdio.h>

#define MAX 25

void worstFit(int b[], int f[], int nb, int nf) {

   int frag[MAX], bf[MAX], ff[MAX];

   for (int i = 0; i < nb; i++) bf[i] = 0;

   for (int i = 0; i < nf; i++) ff[i] = -1;

   for (int i = 0; i < nf; i++) {

      int max_block = -1, max_size = -1;

      for (int j = 0; j < nb; j++) {

         if (bf[j] == 0 && b[j] >= f[i] && b[j] - f[i] > max_size) {

            max_size = b[j] - f[i];

            max_block = j;

         }

      }

      if (max_block != -1) {

         ff[i] = max_block;

         frag[i] = b[max_block] - f[i];

         bf[max_block] = 1;

      } else {

         frag[i] = -1;

      }

   }

   printf("\nWorst Fit Allocation:\n");

   printf("File No\t   File Size\t   Block No\t   Block Size\t   Fragment\n");

   printf("--------------------------------------------------------------\n");

   for (int i = 0; i < nf; i++) {

      if (ff[i] != -1)

         printf("%-10d%-15d%-15d%-15d%-10d\n", i + 1, f[i], ff[i] + 1, b[ff[i]], frag[i]);

      else

         printf("%-10d%-15d%-15s\n", i + 1, f[i], "Not Allocated");
```

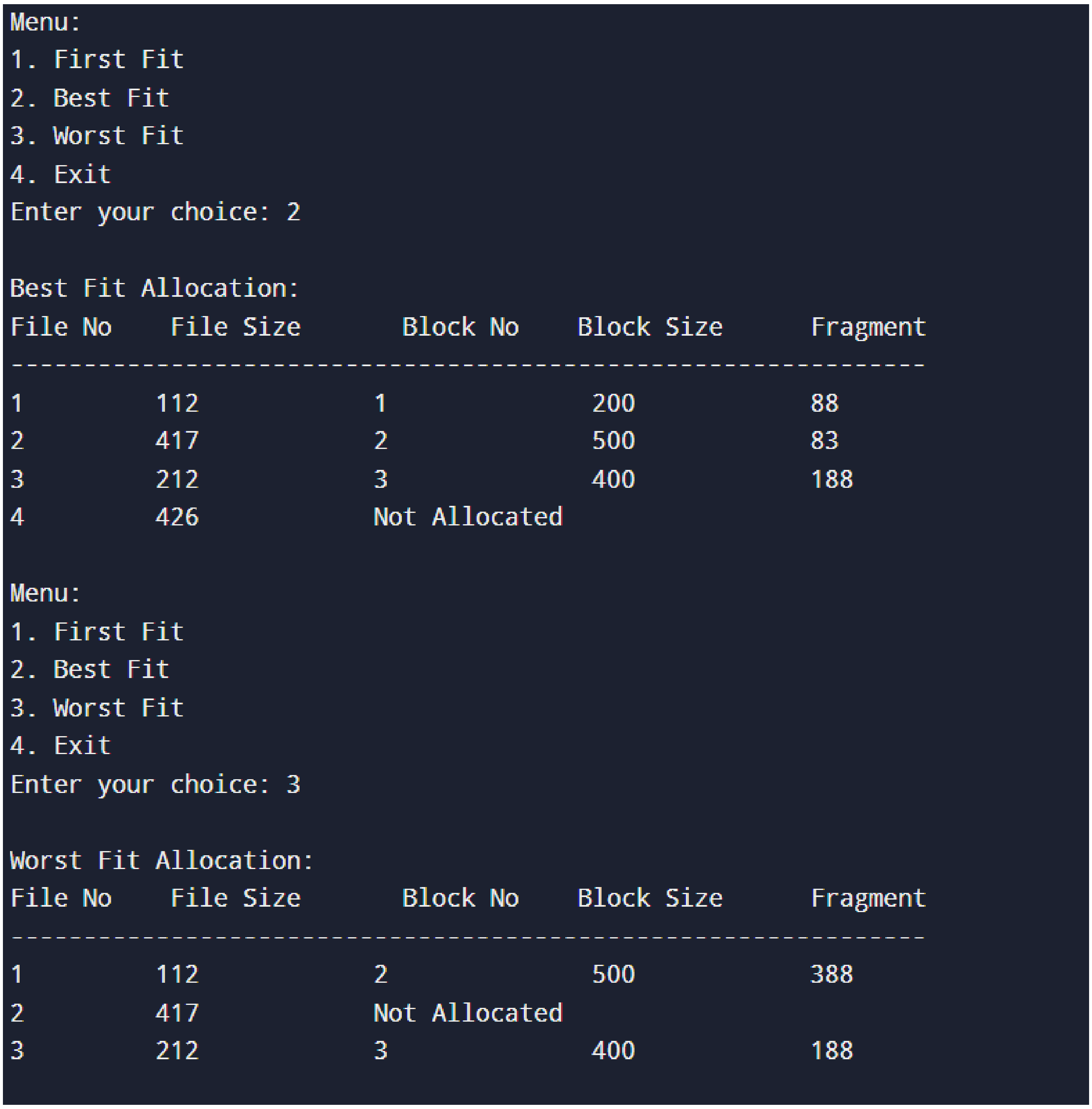
```c
    }
}
void firstFit(int b[], int f[], int nb, int nf) {
    int frag[MAX], bf[MAX], ff[MAX];
    for (int i = 0; i < nb; i++) bf[i] = 0;
    for (int i = 0; i < nf; i++) ff[i] = -1;
    for (int i = 0; i < nf; i++) {
        for (int j = 0; j < nb; j++) {
            if (bf[j] == 0 && b[j] >= f[i]) {
                ff[i] = j;
                frag[i] = b[j] - f[i];
                bf[j] = 1;
                break;
            }
        }
    }
    printf("\nFirst Fit Allocation:\n");
    printf("File No\t   File Size\t   Block No\t   Block Size\t   Fragment\n");
    printf("----------------------------------------------------------------\n");
    for (int i = 0; i < nf; i++) {
        if (ff[i] != -1)
            printf("%-10d%-15d%-15d%-15d%-10d\n", i + 1, f[i], ff[i] + 1, b[ff[i]], frag[i]);
        else
            printf("%-10d%-15d%-15s\n", i + 1, f[i], "Not Allocated");
    }
}
void bestFit(int b[], int f[], int nb, int nf) {
    int frag[MAX], bf[MAX], ff[MAX];
    for (int i = 0; i < nb; i++) bf[i] = 0;
    for (int i = 0; i < nf; i++) ff[i] = -1;
    for (int i = 0; i < nf; i++) {
        int min_block = -1, min_size = 1e9;
        for (int j = 0; j < nb; j++) {
            if (bf[j] == 0 && b[j] >= f[i] && b[j] - f[i] < min_size) {
                min_size = b[j] - f[i];
                min_block = j;
            }
        }
        if (min_block != -1) {
            ff[i] = min_block;
            frag[i] = b[min_block] - f[i];
            bf[min_block] = 1;
        } else {
            frag[i] = -1;
        }
    }
    printf("\nBest Fit Allocation:\n");
    printf("File No\t   File Size\t   Block No\t   Block Size\t   Fragment\n");
    printf("--------------------------------------------------------------\n");
    for (int i = 0; i < nf; i++) {
        if (ff[i] != -1)
            printf("%-10d%-15d%-15d%-15d%-10d\n", i + 1, f[i], ff[i] + 1, b[ff[i]], frag[i]);
        else
            printf("%-10d%-15d%-15s\n", i + 1, f[i], "Not Allocated");
    }
}
int main() {
    int b[MAX], f[MAX], nb, nf, choice;
    printf("\n\tMemory Management Schema \n");
    printf("Enter the number of blocks: ");
    scanf("%d", &nb);
    printf("Enter the sizes of the blocks:\n");
```

```
for (int i = 0; i < nb; i++) {
    printf("Block %d: ", i + 1);
    scanf("%d", &b[i]);
}
printf("Enter the number of files: ");
scanf("%d", &nf);
printf("Enter the sizes of the files:\n");
for (int i = 0; i < nf; i++) {
    printf("File %d: ", i + 1);
    scanf("%d", &f[i]);
}
do {
    printf("\nMenu:\n");
    printf("1. First Fit\n");
    printf("2. Best Fit\n");
    printf("3. Worst Fit\n");
    printf("4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
    switch (choice) {
        case 1:
            firstFit(b, f, nb, nf);
            break;
        case 2:
            bestFit(b, f, nb, nf);
            break;
        case 3:
            worstFit(b, f, nb, nf);
            break;
        case 4:
            printf("Exiting...\n");
            break;
          default:
            printf("Invalid choice. Try again!\n");
        }
    } while (choice != 4);
    return 0;
}
```

Output :



```
    Memory Management Schema
Enter the number of blocks: 3
Enter the sizes of the blocks:
Block 1: 200
Block 2: 500
Block 3: 400
Enter the number of files: 4
Enter the sizes of the files:
File 1: 112
File 2: 417
File 3: 212
File 4: 426

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

First Fit Allocation:
File No    File Size       Block No    Block Size      Fragment
---------------------------------------------------------------
1         112            1              200            88
2         417            2              500            83
3         212            3              400            188
4         426            Not Allocated
```



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

Best Fit Allocation:
File No    File Size      Block No    Block Size      Fragment
--------------------------------------------------------------
1         112            1             200            88
2         417            2             500            83
3         212            3             400            188
4         426            Not Allocated

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

Worst Fit Allocation:
File No    File Size      Block No    Block Size      Fragment
--------------------------------------------------------------
1         112            2             500            388
2         417            Not Allocated
3         212            3             400            188
```