

EX –10 a)

BEST FIT

NAME : Alfred Sam D

CLASS : CSE – A

ROLL NO : 230701026

Program:

```
#include <stdio.h>

#define MAX 25

int main() {
    int blockSize[MAX], processSize[MAX];
    int allocation[MAX];
    int b, p;
    printf("Enter number of memory blocks: ");
    scanf("%d", &b);
    printf("Enter size of each memory block:\n");
    for (int i = 0; i < b; i++) {
        printf("Block %d: ", i + 1);
        scanf("%d", &blockSize[i]);
    }

    printf("Enter number of processes: ");
    scanf("%d", &p);
    printf("Enter size of each process:\n");
    for (int i = 0; i < p; i++) {
        printf("Process %d: ", i + 1);
        scanf("%d", &processSize[i]);
        allocation[i] = -1; // initialize as unallocated
    }
}
```

```

}

for (int i = 0; i < p; i++) {
    int bestIdx = -1;
    for (int j = 0; j < b; j++) {
        if (blockSize[j] >= processSize[i]) {
            if (bestIdx == -1 || blockSize[j] < blockSize[bestIdx]) {
                bestIdx = j;
            }
        }
    }
}

if (bestIdx != -1) {
    allocation[i] = bestIdx + 1; // block number (1-based)
    blockSize[bestIdx] -= processSize[i];
}
}

printf("\nProcess No.\tProcess Size\tBlock No.\n");
for (int i = 0; i < p; i++) {
    printf("%d\t\t%d\t\t", i + 1, processSize[i]);
    if (allocation[i] != -1)
        printf("%d\n", allocation[i]);
    else
        printf("Not Allocated\n");
}

return 0;
}

```

OUTPUT :

```
Enter number of memory blocks: 4
Enter size of each memory block:
Block 1: 150
Block 2: 350
Block 3: 200
Block 4: 100
Enter number of processes: 5
Enter size of each process:
Process 1: 115
Process 2: 500
Process 3: 50
Process 4: 300
Process 5: 200

Process No.    Process Size    Block No.
1             115            1
2             500           Not Allocated
3             50            4
4             300            2
5             200            3
-bash-4.4$
```



10 b)

FIRST FIT

Program :

```
#include <stdio.h>

#define MAX 25

int main() {

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

    int i, j, nb, nf;

    printf("Enter the number of blocks: ");

    scanf("%d", &nb);

    printf("Enter the number of files: ");

    scanf("%d", &nf);


    printf("Enter the size of each block:\n");

    for (i = 0; i < nb; i++) {

        printf("Block %d: ", i + 1);

        scanf("%d", &b[i]);

        bf[i] = 0; // mark all blocks as unallocated

    }


    printf("Enter the size of each file:\n");

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

        printf("File %d: ", i + 1);

        scanf("%d", &f[i]);

    }

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

        for (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; // mark block as allocated

        break;

    }

}

if (j == nb) {

    ff[i] = -1; // file not allocated

    frag[i] = -1;

}

}

printf("\nFile No\tFile Size\tBlock No\tBlock Size\tFragment\n");

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

    printf("%d\t%d\t\t", i + 1, f[i]);

    if (ff[i] != -1)

        printf("%d\t\t%d\t\t%d\n", ff[i] + 1, b[ff[i]], frag[i]);

    else

        printf("Not Allocated\n");

}

return 0;

}

```

OUTPUT :

```
-bash-4.4$ vi first_fit.c
-bash-4.4$ gcc first_fit.c
-bash-4.4$ ./a.out
Enter the number of blocks: 4
Enter the number of files: 3
Enter the size of each block:
Block 1: 5
Block 2: 0
Block 3: 4
Block 4: 10
Enter the size of each file:
File 1: 1
File 2: 4
File 3: 7
```

File No	File Size	Block No	Block Size	Fragment
1	1	1	5	4
2	4	3	4	0
3	7	4	10	3

```
-bash-4.4$
```

