```
//PROGRAM OF BEST FIT
#include <stdio.h>
void implimentBestFit(int blockSize[], int blocks, int processSize[], int
proccesses)
{

    int allocation[proccesses];
    int occupied[blocks];


    for(int i = 0; i < proccesses; i++){
        allocation[i] = -1;
    }

    for(int i = 0; i < blocks; i++){
        occupied[i] = 0;
    }


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

        int indexPlaced = -1;
        for (int j = 0; j < blocks; j++) {
            if (blockSize[j] >= processSize[i] && !occupied[j])
            {

                if (indexPlaced == -1)
                    indexPlaced = j;


                else if (blockSize[j] < blockSize[indexPlaced])
                    indexPlaced = j;
            }
        }
        if (indexPlaced != -1)
        {

            allocation[i] = indexPlaced;


            occupied[indexPlaced] = 1;
        }
    }

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



int main()
{
    int blockSize[] = {100, 50, 30, 120, 35};
    int processSize[] = {40, 10, 30, 60};
    int blocks = sizeof(blockSize)/sizeof(blockSize[0]);
    int proccesses = sizeof(processSize)/sizeof(processSize[0]);

    implimentBestFit(blockSize, blocks, processSize, proccesses);

    return 0 ;
}>= processSize[i] && !occupied[j])
            {

                if (indexPlaced == -1)
```

```
                    indexPlaced = j;


                else if (blockSize[j] < blockSize[indexPlaced])
                    indexPlaced = j;
            }
        }

        if (indexPlaced != -1)
        {

            allocation[i] = indexPlaced;


            occupied[indexPlaced] = 1;
        }
    }

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


int main()
{
    int blockSize[] = {100, 50, 30, 120, 35};
    int processSize[] = {40, 10, 30, 60};
    int blocks = sizeof(blockSize)/sizeof(blockSize[0]);
    int proccesses = sizeof(processSize)/sizeof(processSize[0]);
    implimentBestFit(blockSize, blocks, processSize, proccesses);

    return 0 ;
}
```

OUTUT

```
Process No.   Process Size   Block no.
1             40             2
2             10             3
3             30             5
4             60             1
```