

PROGRAM CODE

```
#include<stdio.h>
#include<stdlib.h>

void first_fit(int n,int m,int blo[],int proc[])
{
    int alloc[n];
    for(int i=0;i<n;i++)
    {
        alloc[i]=-1;
    }
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(alloc[j]==-1&&proc[i]<=blo[j])
            {
                alloc[j]=proc[i];
                break;
            }
        }
    }
    for(int i=0;i<n;i++)
    {
        if(alloc[i]!=-1)
        {
            printf("\n%d is allocated at block
%d\n",alloc[i],blo[i]);
        }
        else
        {
            printf("\n%d block is not allocated\n",blo[i]);
        }
    }
}

void worst_fit(int n,int m,int blo[],int proc[])
{
    int alloc[n];
    for(int i=0;i<n;i++)
    {
        alloc[i]=-1;
    }
    int max=blo[0],k=0;
    for(int i=0;i<m;i++)
    {
        max=0;
        for(int j=0;j<n;j++)
```

```

        {
            if(max<blo[j]&&alloc[j]==-1)
            {
                max=blo[j];
                k=j;
            }
        }
        if(proc[i]<=max&&alloc[k]==-1)
        {
            alloc[k]=proc[i];
        }
    }
    for(int i=0;i<n;i++)
    {
        if(alloc[i]!=-1)
        {
            printf("\n%d is allocated at block
%d\n",alloc[i],blo[i]);
        }
        else
        {
            printf("\n%d block is not allocated\n",blo[i]);
        }
    }
}
void best_fit(int n,int m,int blo[],int proc[])
{
    int alloc[n];
    for(int i=0;i<n;i++)
    {
        alloc[i]=-1;
    }
    int temp=10000,k=0;
    for(int i=0;i<m;i++)
    {
        temp=10000;
        for(int j=0;j<n;j++)
        {
            if(proc[i]<=blo[j]&&alloc[j]==-1&&temp>blo[j]-proc[i])
            {
                temp=blo[j]-proc[i];
                k=j;
            }
        }
        if(alloc[k]==-1)
        {

```

```

        alloc[k]=proc[i];
    }
}
for(int i=0;i<n;i++)
{
    if(alloc[i]!=-1)
    {
        printf("\n%d is allocated at block
%d\n",alloc[i],blo[i]);
    }
    else
    {
        printf("\n%d block is not allocated\n",blo[i]);
    }
}
}
void main()
{
    int n;
    printf("\nenter the number of blocks\n");
    scanf("%d",&n);
    int blo[n];
    int alloc[n];
    printf("\nenter the size of each block\n");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&blo[i]);
    }
    printf("\nenter the number of processors\n");
    int m;
    scanf("%d",&m);
    int proc[m];
    printf("\nenter the processor size\n");
    for(int i=0;i<m;i++)
    {
        scanf("%d",&proc[i]);
    }
    printf("\nworst fit allocations\n");
    worst_fit(n,m,blo,proc);
    printf("\nfirst fit allocations\n");
    first_fit(n,m,blo,proc);
    printf("\nbest fit allocations\n");
    best_fit(n,m,blo,proc);
}

```

OUTPUT

```
sahal@kali:~/bash_script$ ./a.out
```

```
enter the number of blocks
```

```
5
```

```
enter the size of each block
```

```
212
```

```
300
```

```
450
```

```
70
```

```
150
```

```
enter the number of processors
```

```
4
```

```
enter the processor size
```

```
215
```

```
500
```

```
60
```

```
260
```

```
worst fit allocations
```

```
212 block is not allocated
```

```
60 is allocated at block 300
```

```
215 is allocated at block 450
```

```
70 block is not allocated
```

```
150 block is not allocated
```

```
first fit allocations
```

```
60 is allocated at block 212
```

```
215 is allocated at block 300
```

```
260 is allocated at block 450
```

```
70 block is not allocated
```

```
150 block is not allocated
```

```
best fit allocations
```

```
212 block is not allocated
```

215 is allocated at block 300

260 is allocated at block 450

60 is allocated at block 70

150 block is not allocated