

NAME:LAIBA NADEEM

ROLL NUMBER:DT-22028

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

```
main()
```

```
{  
    int p[10], b[10], cb[10], flag[10];  
    int i, j, np, nb;  
    printf("Enter the no of processes:");  
    scanf("%d",&np);  
    printf("Enter the no of blocks:");  
    scanf("%d",&nb);  
    printf("Enter the size of each process:");  
    for(i=0;i<np;i++)  
    {  
        printf("\nProcess %d:",i+1);  
        scanf("%d",&p[i]);  
    }  
}
```

```
printf("\nEnter the block sizes:");  
for(j=0;j<nb;j++)  
{  
    printf("\nBlock %d:",j+1);  
    scanf("%d",&b[j]);  
}
```

```
if(np>nb)  
    printf("\nFirst fit 2: Best fit 3: Worst fit:");  
for(i=0;i<np;i++)  
{  
    printf("\nEnter your choice:");  
    scanf("%d",&cb[i]);  
    switch(cb[i])  
    {  
        case 1: printf("\nFirst Fit");  
                for(i=0;i<np;i++)
```

```

{
    for(j=0;j<nb;j++)
    {
        if(p[i]<=b[j])
        {
            printf("\nProcess %d of size %d is allocated in block of size %d",i+1,p[i],b[j]);
            flag[i]=1;
            b[j]-=p[i];
            break;
        }
    }
    if(flag[i]==0)
        printf("\nProcess %d of size %d is not allocated",i+1,p[i]);
}
break;
case 2: printf("\nBest Fit");
for(i=0;i<np;i++)
{
    for(j=i+1;j<nb;j++)
    {
        if(b[i]>b[j])
        {
            int temp;
            temp=b[i];
            b[i]=b[j];
            b[j]=temp;
        }
    }
}
printf("\nAfter sorting block sizes:");
for(i=0;i<nb;i++)
{
    printf("\nBlock %d:%d",i+1,b[i]);
}
break;
}
}
}

```

```

if(p[i] <= c[j])
{
    alloc[j] = p[i];
    printf("\n\nAlloc[%d]", alloc[j]);
    printf("\n\nProcess %d of size %d is allocated in block %d of size %d", i, p[i], j, c[j]);
    flag[i] = 0;
    c[j] = 0;
    break;
}
else
    flag[i] = 1;
}

```

```

for(i = 0; i < np; i++)
{
    if(flag[i] != 0)
        printf("\n\nProcess %d of size %d is not allocated", i, p[i]);
}

```

```

break;
case 3:
    printf("\nWorst Fit\n");
    for(i = 0; i < nb; i++)
    {
        for(j = i + 1; j < nb; j++)
        {
            if(d[i] < d[j])
            {
                int temp = d[i];
                d[i] = d[j];
                d[j] = temp;
            }
        }
    }
}

```

```

d[i] = d[j];
d[j] = temp;

```

```

}
}
printf("\nAfter sorting block sizes:");
for(i = 0; i < nb; i++)
    printf("\nBlock %d:%d", i, d[i]);

for(i = 0; i < np; i++) {
    for(j = 0; j < nb; j++) {
        if(p[i] <= d[j]) {
            alloc[j] = p[i];
            printf("\n\nAlloc[%d]", alloc[j]);
            printf("\n\nProcess %d of size %d is allocated in block %d of size %d", i, p[i], j, d[j]);

            flag[i] = 0;
            d[j] = 0;
            break;
        } else {
            flag[i] = 1;
        }
    }
}

for(i = 0; i < np; i++) {
    if(flag[i] != 0)
        printf("\n\nProcess %d of size %d is not allocated", i, p[i]);
}

break;

default:
    printf("Invalid Choice...!");
    break;
}

} while(ch <= 3);
}

```

1. First Fit 2. Best Fit 3. Worst Fit

First Fit

Process 0 of size 100 is allocated in block 0 of size 500

Process 2 of size 200 is allocated in block 1 of size 400

Process 3 of size 300 is allocated in block 2 of size 300

Process 1 of size 500 is not allocated

Process 4 of size 600 is not allocated

Best Fit

After sorting block sizes:

Block 0: 100

Block 1: 200

Block 2: 300

Block 3: 400

Block 4: 500

Process 0 of size 100 is allocated in block 0 of size 100

Process 1 of size 500 is allocated in block 4 of size 500

Process 2 of size 200 is allocated in block 1 of size 200

Process 3 of size 300 is allocated in block 2 of size 300

Process 4 of size 600 is not allocated

Worst Fit

After sorting block sizes:

Block 0: 500

Block 1: 400

Block 2: 300

Block 3: 200

Block 4: 100

Process 0 of size 100 is allocated in block 0 of size 500

Process 2 of size 200 is allocated in block 1 of size 400

Process 3 of size 300 is allocated in block 2 of size 300

Process 1 of size 500 is not allocated

Process 4 of size 600 is not allocated

