

Write a program to simulate Memory placement strategies – best fit, first fit, next fit and worst fit.

operating system must keep list of each memory location noting **which are free and which are busy.**

Then as new jobs come into the system, the free partitions must be allocated. These partitions may be allocated by 4 ways:

1. First-Fit Memory Allocation
2. Best-Fit Memory Allocation
3. Worst-Fit Memory Allocation
4. Next-Fit Memory Allocation

These are **Contiguous** memory allocation techniques.

First Fit

In the first fit approach is to **allocate the first free partition or hole large enough which can accommodate the process**. It finishes after finding the first suitable free partition.

Best Fit

The best fit deals with **allocating the smallest free partition** which meets the requirement of the requesting process. This algorithm first searches the entire list of free partitions and considers the smallest hole that is adequate. It then tries to find a hole which is close to actual process size needed.

Worst fit

In worst fit approach is to **locate largest available free portion** so that the portion left will be big enough to be useful. It is the reverse of best fit

first fit

In the first fit approach is to allocate the first free partition or hole large enough which can accommodate the process. It finishes after finding the first suitable free partition.

Enter the number of blocks:4

Enter the number of files:3

Enter the size of the blocks:-

Block 1:5

Block 2:8

Block 3:4

Block 4:10

Enter the size of the files:-

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	2	8	4
3	7	4	10	3_

best fit

the best fit deals with allocating the **smallest free partition** which meets the requirement of the requesting process.

This algorithm first searches the entire list of free partitions and considers the smallest hole that is adequate..

```
Enter the number of blocks:4
```

```
Enter the number of files:3
```

```
Enter the size of the blocks:-
```

```
Block 1:5
```

```
Block 2:8
```

```
Block 3:4
```

```
Block 4:10
```

```
Enter the size of the files:-
```

```
File 1:1
```

```
File 2:4
```

```
File 3:7
```

File_no	File_size	Block_no	Block_size	Fragment
1	1	3	4	3
2	4	1	5	1
3	7	2	8	1

In worst fit approach is to **locate largest available** free portion so that the portion left will be big enough to be useful. It is the reverse of best fit.

```
Enter the number of blocks:4
Enter the number of files:3

Enter the size of the blocks:-
Block 1:5
Block 2:8
Block 3:4
Block 4:10
Enter the size of the files:-
File 1:1
File 2:4
File 3:7
```

File_no	File_size	Block_no	Block_size	Fragment
1	1	4	10	9
2	4	2	8	4
3	7	0	0	0

Next fit is a modified version of first fit. **It begins as first fit to find a free partition. When called next time it starts searching from where it left off, not from the beginning.**

What is Next Fit ?

Next fit is a modified version of first fit. **It begins as first fit to find a free partition. When called next time it starts searching from where it left off**, not from the beginning

Example:

Input : blockSize[] = {5, 10, 20};

processSize[] = {10, 20, 30};

Output:

Process No.	Process Size	Block no.
1	10	2
2	20	3
3	30	Not Allocated