Experiment No.: 4

Name : Khushi Chaudhari

Roll No.: 07

Batch : T5

Problem Statement : Write a Java/C++ program to simulate memory placement strategies
1. First Fit
2. Best Fit
3. Worst Fit

Code :

1. WORST-FIT :

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

#define max 25

void main()

{ int frag[max],b[max],f[max],i,j,nb,nf, temp;

static int bf[max],ff[max];

printf("\n\tMemory Management Scheme - First Fit");

clrscr();

printf("\nEnter the number of blocks:"); ;

scanf("%d", &nb)
```

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

scanf("%d", &nf);

printf("\nEnter the size of the blocks:-\n"); for(i=1;i<=nb;i++)

{ printf("Block %d:",i); scanf("%d",&b[i]);

} printf("Enter the size of the files :-\n"); for (i=1; i <= nf ,i++) {

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

for(i=1;i<=nf;i++) {- for(j=1;j<=nb;j++) ( if(bf[j] * I = 1) { temp=b[j]-f[i]; f (temp>=0)

ff[1] = j break;

}

}

}

frag[i]=temp; bf[ff[i]] = 1

} printf("\nFile_no:\tFile_size:\tBlock_no:\tBlock_size:\tFragement"); for ( i = 1 ,i<= cap f;i++)

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

getch();
```

}

INPUT

Enter the number of blocks: 3

Enter the number of files: 2

Enter the size of the blocks:-

Block 1: 5

Block 2: 2

Block 3: 7

Enter the size of the files:

File 1:1

File 2:4

OUTPUT

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

2. BEST-FIT:

#include<stdio.h>

```
#include<conio.h>

#define max 25

void main()

{ int frag[max],b[max],f[max],i,j,nb,nf, temp,lowest t = 10000

static int bf[max],ff[max];

clrscr();

printf("\nEnter the number of blocks:");

scanf("%d",&nb);

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

scanf("%d",&nf);

printf("\nEnter the size of the blocks:-\n"); for ( i = 1; i <= nb ;i++)

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

printf("Enter the size of the files :-\n"); for ( i = 1; i <= nf ;i++)

{

printf("File %d:",i);
```

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

} for( ( i = 1; i <= nf ;i++) {

for ( j = 1; j <= nb ;j++)

{ if(bf * [j]! = 1)

{ temp=b[j]-f[i]; if (temp>=0) if(lowest>temp) {

ff[i] = j

lowest=temp;
}

}

}

frag[i]=lowest;

bf[ff[i]]=1;

lowest=10000;

}

printf("\nFile No\tFile Size \tBlock No\tBlock Size\tFragment"); for(i=1;i<=nf && ff[i]!=0;i++)
```

```
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i], ff[i], b[ff[i]], frag[i]); getch();

}
```

INPUT

```
Enter the number of blocks: 3

Enter the number of files: 2

Enter the size of the blocks:-

Block 1: 5

Block 2: 2

Block 3: 7

Enter the size of the files:-

File 1: 1

File 2: 4
```

OUTPUT

```
File No File Size Block No Block Size Fragment

11221
```

24151

3. FIRST-FIT:

```
#include<stdio.h>

#include<conio.h>

#define max 25
```

```
void main()

{

int frag[max],b[max],f[max],i,j,nb,nf, temp,highest=0; static int bf[max], ff[max]; clrscr();

printf("\n\tMemory Management Scheme - Worst Fit");

printf("\nEnter the number of blocks:");

scanf("%d",&nb);

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

scanf("%d",&nf);

printf("\nEnter the size of the blocks:-\n"); for(i=1;i<=nb;i++)

{

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

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

}

printf("Enter the size of the files :-\n"); for(i=1;i<=nf;i++) {
printf("File %d:", i);
```

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

} yr( i = 1; i <= nf ;i+++)

{ r( j = 1; j <= nb ;j++)

{ if(bf[j] * l = 1) //if bf[j] not allocated

{

temp=b[j]-f[i];

if (temp>=0)

if(highest<temp)

{

ff[i]=j;

highest=temp;

}

}

} frag[i]=highest;

bf[ff[i] |= 1
```

```
highest = 0 ;

} printf("\nFile_no:\tFile_size:\tBlock_no:\tBlock_size:\tFragement"); for (i=1; i <= nf ,i++)

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

getch();

}
```

INPUT

```
Enter the number of blocks: 3

Enter the number of files: 2

Enter the size of the blocks:-

Block 1: 5

Block 2: 2

Block 3: 7

Enter the size of the files:-

File 1: 1

File 2: 4
```

OUTPUT

File No File Size Block No Block Size Fragment

11376

24151