NAME- Neelam Gusain

COURSE- BSc IT(A)

STUDENT Id- 20052049

Write a c program to implement Best Fit memory management algorithm.

ALGORITHM

1. START
2. Get no. of Processes and no. of blocks.
3. After that get the size of each block and process request.
4. Then select the best memory block that can be allocated.
5. Display the processes with the blocks that are allocated to a respective process.
6. Value of Fragmentation is optional to display to keep track of wasted memory.
7. STOP

CODE

```
#include<stdio.h>
void main ()
{
 int fragment [20], b [20], p [20], i, j, nb, np, temp, lowest=9999;
 static int barray [20], parray [20];
 printf (“\n\t\t\t Memory Management Scheme – Best Fit”);
 printf (“\n Enter the number of blocks:”);
 scanf (“%d”, & nb);
 printf (“Enter the number of processes:”);
 scanf (“%d”, & np);
printf (“\n Enter the size of the blocks: - \n”);
for (i=1; i<= nb; i++)
{
 printf (“Block no.%d:”, i);
 scanf (“%d”, &b[i]);
}
printf (“\n Enter the size of the processes: - \n”);
for (i=1; i<= np; i++)
{
 printf (“Process no. %d:”, i);
 scanf (“%d”, &p[i]);
}
for(i=1; i<= np; i++)
{
for(j=1; j<= nb; j++)
{
if (barray[j]! = 1)
{
```

```
temp= b[j]- p[i];
if(temp>= 0)
if(lowest>temp)
{
parray[i]= j;
lowest= temp;
}
}
}
fragment [i]= lowest;
barray[parray[i]]= 1;
lowest= 10000;
}
printf (“\n Process_no\t Process_size\t Block_no\t Block_size\t
Fragment”);
for (i=1; i<= np && parray[i]! =0; i++)
printf (“\n%d\t\t%d\t\t%d\t\t%d\t\t%d”, i, p[i], parray[i],
b[parray[i]], fragment[i]);
}
```

OUTPUT

```
Enter the number of blocks:5
Enter the number of processes:4

Enter the size of the blocks:-
Block no.1:10
Block no.2:15
Block no.3:5
Block no.4:9
Block no.5:3
Enter the size of the processes :-
Process no.1:1
Process no.2:4
Process no.3:7
Process no.4:12

Process_no      Process_size    Block_no        Block_size      Fragment
1               1               5               3               2
2               4               3               5               1
3               7               4               9               2
4               12              2               15              3
Process returned 4 (0x4)   execution time : 33.196 s
Press any key to continue.
```

Write a c program to implement Worst Fit memory management algorithm.

ALGORITHM

1. START
2. Get no. of Processes and no. of blocks.
3. After that get the size of each block and process request.
4. Then select the worst memory block that can be allocated.
5. Display the processes with the blocks that are allocated to a respective process.
6. STOP

CODE

```
#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\t Memory Management Scheme- Worst Fit”);
 printf (“\n Enter the number of blocks:”);
 scanf (“%d”, &nb);
 printf (“Enter the number of files:”);
 scanf (“%d”, &nf);
 printf (“\n Enter 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 (highest<temp)
 {
  ff[i]= j;
  highest= temp;
 }
}
}
frag[i]= highest;
bf[ff[i]]=1;
highest= 0;
}
printf (“\n File_no:\t File_size:\t Block_no:\t Block_size:\t
Fragment”);
For (i=1; i<=n; 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();
}
```

OUTPUT


Memory Management Scheme - Worst Fit
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
File_no: File_size : Block_no: Block_size: Fragment
1 1 3 7 6
2 4 1 5 1