. WAP to simulate the following contiguous memory allocation technique:

a. Worst-fit b. Best-fit c. First-fit

**a. Memory Allocation (Worst fit)**

**Code:**

#include<stdio.h>

#define max 25

int main() {

int frag[max], b[max], f[max], bf[max], ff[max];

int i, j, nb, nf, temp;

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 = 0; i < nb; i++) {

printf("Block %d:", i + 1);

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

}

printf("Enter the size of the files :-\n");

for(i = 0; i < nf; i++) {

printf("File %d:", i + 1);

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

}

for(i = 0; i < nf; i++) {

ff[i] = -1; // Initialize to -1 indicating no block assigned

frag[i] = -1; // Initialize to -1 indicating not allocated

}

for(i = 0; i < nf; i++) {

temp = -1;

for(j = 0; j < nb; j++) {

if(bf[j] != 1) { // If block is not allocated

if(b[j] >= f[i]) { // If file fits in the block

if(temp == -1 || b[temp] < b[j]) {

temp = j; // Update temp to point to the largest block

}

}

}

}

if(temp != -1) {

ff[i] = temp;

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

bf[temp] = 1; // Mark the block as allocated

}

}

printf("\nFile\_no:\tFile\_size :\tBlock\_no:\tBlock\_size:\tFragmentation\n");

for(i = 0; i < nf; i++) {

printf("%d\t\t%d\t\t", i + 1, f[i]);

if(ff[i] != -1) {

printf("%d\t\t%d\t\t%d\n", ff[i] + 1, b[ff[i]], frag[i]);

} else {

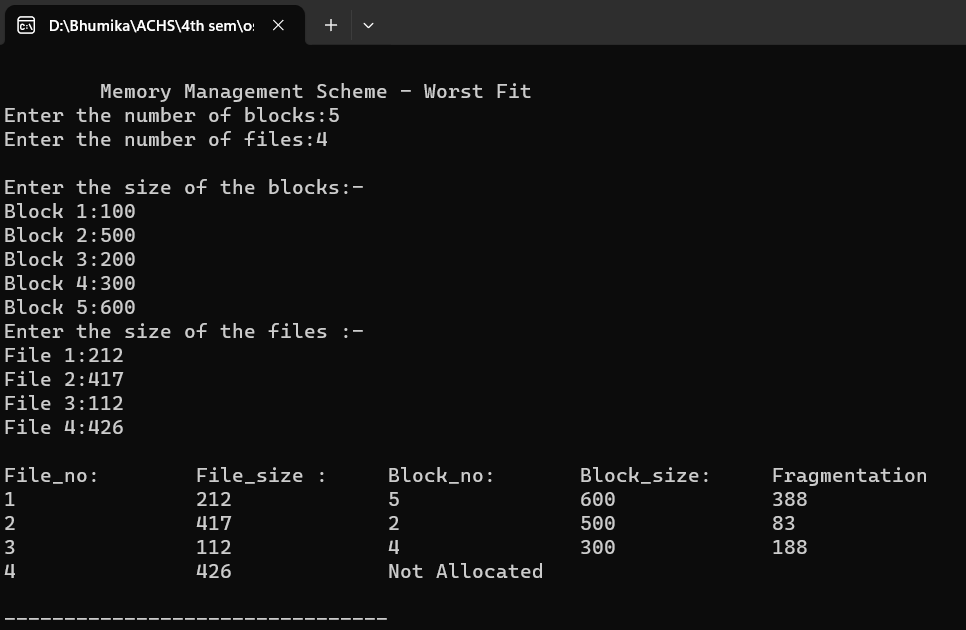
printf("Not Allocated\n");

}

}

return 0;

}**Output:**

****

**b. Memory Allocation (Best fit)**

#include<stdio.h>

#include<conio.h>

#define max 25

int main()

{

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

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

printf("\n\tMemory Management Scheme - Best 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]);

}

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;

}

}}

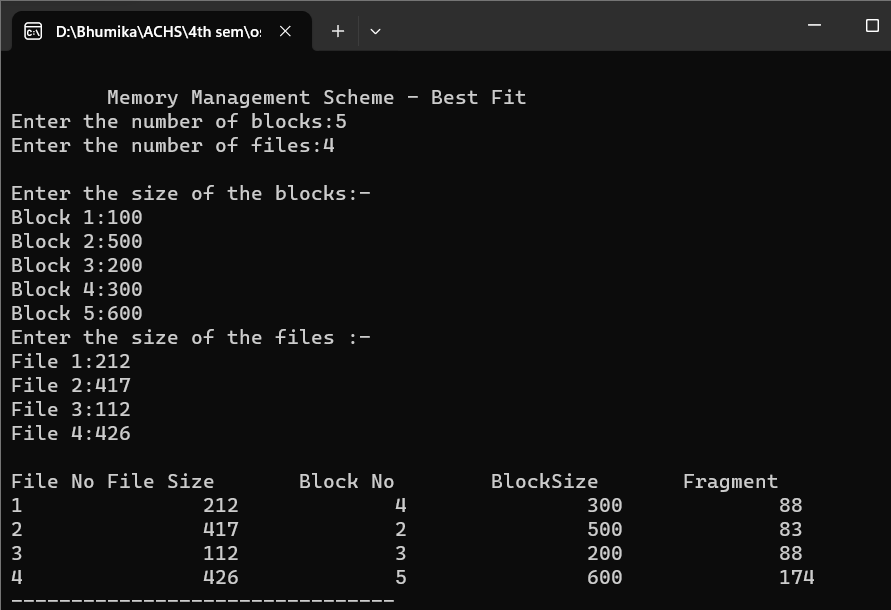
frag[i]=lowest; bf[ff[i]]=1; lowest=10000;

}

printf("\nFile No\tFile Size \tBlock No\tBlockSize\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]); }

**Output:**

****

**c. Memory Allocation (First fit)**

#include<stdio.h>

#include<conio.h>

int main()

{

int bsize[10], psize[10], bno, pno, flags[10], allocation[10], i, j; for(i = 0; i < 10; i++)

{

flags[i] = 0;

allocation[i] = -1;

}

printf("Enter no. of blocks: ");

scanf("%d", &bno);

printf("\nEnter size of each block: ");

for(i = 0; i < bno; i++)

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

printf("\nEnter no. of processes: ");

scanf("%d", &pno);

printf("\nEnter size of each process: ");

for(i = 0; i < pno; i++)

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

for(i = 0; i < pno; i++) //allocation as per first fit

for(j = 0; j < bno; j++)

if(flags[j] == 0 && bsize[j] >= psize[i])

{

allocation[j] = i;

flags[j] = 1;

break;

}

//display allocation details

printf("\nBlock no.\tsize\t\tprocess no.\t\tsize");

for(i = 0; i < bno; i++)

{

printf("\n%d\t\t%d\t\t", i+1, bsize[i]);

if(flags[i] == 1)

printf("%d\t\t\t%d",allocation[i]+1,psize[allocation[i]]);

else

printf("Not allocated");

}

}

**Output:**

