**Program 11:** Write a program to implement the worst fit memory management algorithm. The program should take input total no. of the memory block, their sizes, process name, and process size. The output of the program should give the details about memory allocated to process with fragmentation detail.

**Answer:**

Source Code:

#include<stdio.h>

#include<stdlib.h>

typedef struct{

char process\_name[3];

int size,allocated;

}process;

typedef struct{

int size,fragment\_size,allocated;

}mem;

void algorithm(mem mem\_block[],int n, process pr[], int m){

int i,j,ind=-1;

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

ind=-1;

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

if(mem\_block[j].allocated==0){

if(ind==-1){ //check if ind is alloted

ind=j;

}

if(mem\_block[j].size>=mem\_block[ind].size){ // finding biggest mem block

ind=j;

}

}

}

if(mem\_block[ind].size>=pr[i].size){

mem\_block[ind].fragment\_size=mem\_block[ind].size-pr[i].size;

pr[i].allocated=ind;

mem\_block[ind].allocated=1;

}

}

}

void print\_table(process pr[],int m, mem mem\_block[]){

int i,frag;

puts(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

puts("| Process name | Size | Alloted | Fragment |");

puts("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|");

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

if(pr[i].allocated==-1)

frag =-1;

else

frag=mem\_block[pr[i].allocated].fragment\_size;

printf("| %s | %3d | %2d | %3d |\n",

pr[i].process\_name,pr[i].size,pr[i].allocated,frag);

}

puts("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|");

}

void main(){

int n,m,i,j;

printf("Enter total number of memory blocks\t");

scanf("%d",&n);

mem mem\_block[n];

printf("Enter the block sizes\n");

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

scanf("%d",&mem\_block[i].size);

mem\_block[i].fragment\_size=0;

mem\_block[i].allocated=0;

}

printf("Enter total number of processes\t");

scanf("%d",&m);

process pr[m];

printf("Enter process details--> Process Name, Process Size.\n");

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

scanf("%s %d",pr[i].process\_name,&pr[i].size);

pr[i].allocated=-1;

}

algorithm(mem\_block,n,pr,m);

print\_table(pr,m,mem\_block);

}

Output:

