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: