<u>Problem 10:</u> Write a program to implement the Best fit memory management algorithm. Program should take input total no. of memory block, their sizes, process name and process size. Output of program should give the details about memory allocated to process with fragmentation detail.

Answer:

Source code:

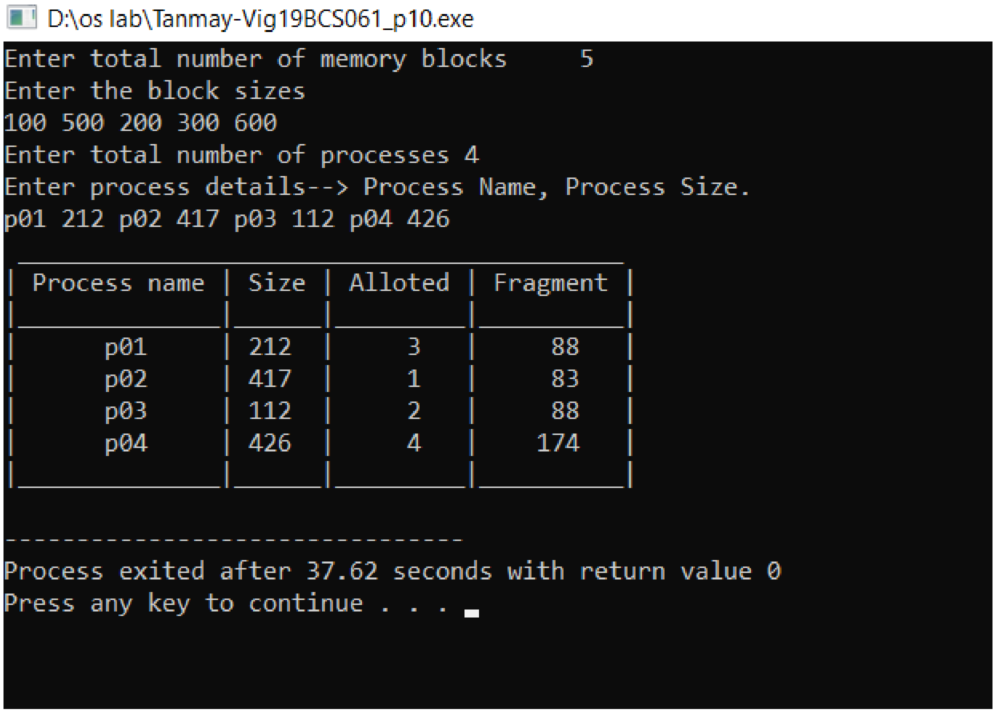
```
#include<stdio.h>

#include<stdlib.h>


typedef struct{

   char process_name[3];

   int size,allocated;

}process;


typedef struct{

   int size,fragment_size;

}mem;


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

   int i,j,best_block=-1;
```

```
    for(i=0;i<m;i++){ // iterate in process array

        best_block=-1;

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

            if(mem_block[j].fragment_size==0 && mem_block[j].size>=pr[i].size){

                if(best_block==-1){

                    best_block=j;

                }

                else if(mem_block[best_block].size>=mem_block[j].size){

                    best_block=j;

                }

            }

        }

        pr[i].allocated=best_block;

        mem_block[best_block].fragment_size=mem_block[best_block].size-pr[i].size;

    }

}



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;
  }
  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:__



```
Enter total number of memory blocks     5
Enter the block sizes
100 500 200 300 600
Enter total number of processes 4
Enter process details--> Process Name, Process Size.
p01 212 p02 417 p03 112 p04 426
 __________________________________________
| Process name | Size | Alloted | Fragment |
|______________|______|_________|__________|
|     p01      | 212  |    3    |    88    |
|     p02      | 417  |    1    |    83    |
|     p03      | 112  |    2    |    88    |
|     p04      | 426  |    4    |   174    |
|______________|______|_________|__________|

--------------------------------
Process exited after 37.62 seconds with return value 0
Press any key to continue . . .
```