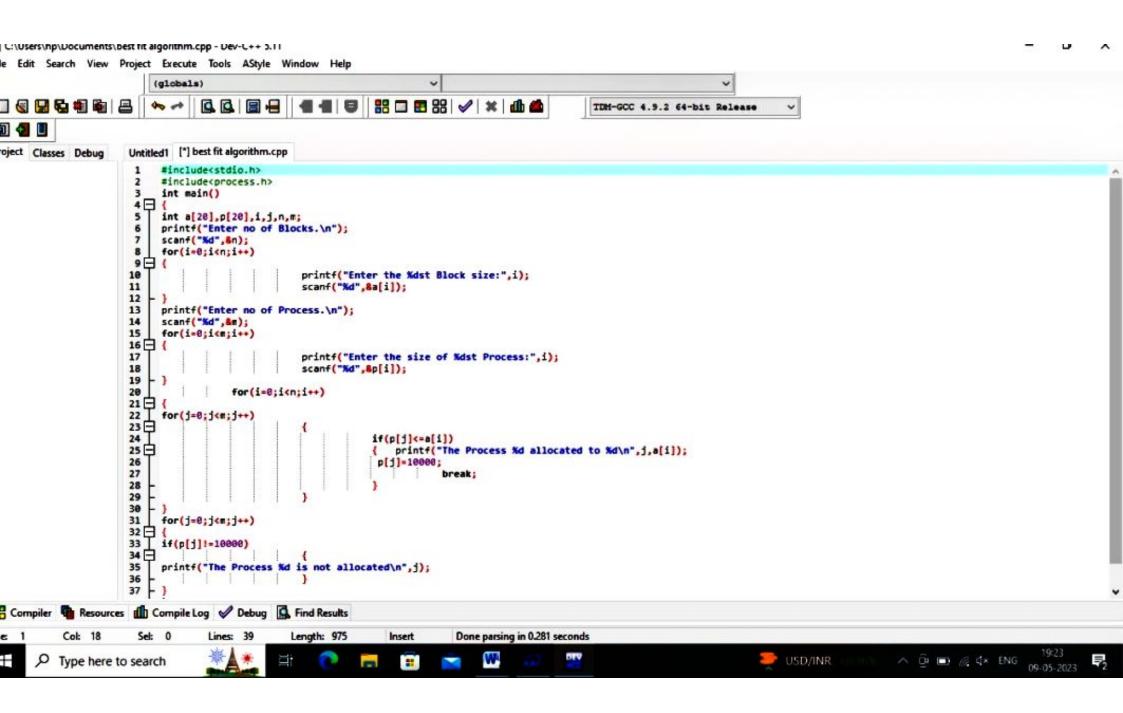12. Write a C program to implement the best-fit algorithm and allocate the memory block to each process.

Test Case:

Memory partitions: 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order),

Show the outcome for the test case with the best-fit algorithms to place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)

(globals)

TDM-GCC 4.9.2 64-bit Release

Untitled1   [*] best fit algorithm.cpp

```c
#include<stdio.h>
#include<process.h>
int main()
{
int a[20],p[20],i,j,n,m;
printf("Enter no of Blocks.\n");
scanf("%d",&n);
for(i=0;i<n;i++)
{
                        printf("Enter the %dst Block size:",i);
                        scanf("%d",&a[i]);
}
printf("Enter no of Process.\n");
scanf("%d",&m);
for(i=0;i<m;i++)
{
                        printf("Enter the size of %dst Process:",i);
                        scanf("%d",&p[i]);
}
            for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
            {
                        if(p[j]<=a[i])
                        {   printf("The Process %d allocated to %d\n",j,a[i]);
                         p[j]=10000;
                                    break;
                        }
            }
}
for(j=0;j<m;j++)
{
if(p[j]!=10000)
            {
printf("The Process %d is not allocated\n",j);
            }
}
```

Compiler Resources Compile Log Debug Find Results

e 1   Col: 18   Sel: 0   Lines: 39   Length: 975   Insert   Done parsing in 0.281 seconds

```
Enter no of Blocks.
6
Enter the 0st Block size:300
Enter the 1st Block size:600
Enter the 2st Block size:350
Enter the 3st Block size:200
Enter the 4st Block size:750
Enter the 5st Block size:125
Enter no of Process.
5
Enter the size of 0st Process:115
Enter the size of 1st Process:500
Enter the size of 2st Process:358
Enter the size of 3st Process:200
Enter the size of 4st Process:375
The Process 0 allocated to 300
The Process 1 allocated to 600
The Process 3 allocated to 350
The Process 2 allocated to 750
The Process 4 is not allocated


--------------------------------
Process exited after 79.48 seconds with return value 0
Press any key to continue . . . _
```