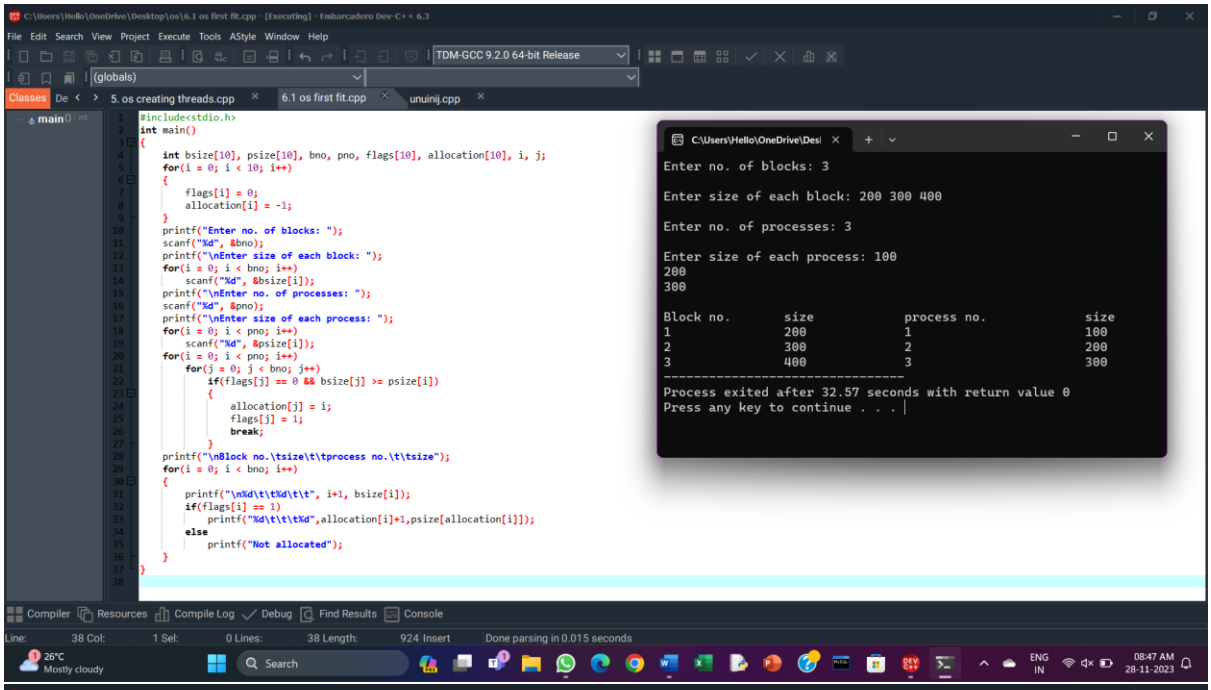


```
#include<stdio.h>
int main()
{
    int bsize[10], psize[10], bno, pno, flags[10], allocation[10], i, j;
    for(i = 0; i < 10; i++)
    {
        flags[i] = 0;
        allocation[i] = -1;
    }
    printf("Enter no. of blocks: ");
    scanf("%d", &bno);
    printf("\nEnter size of each block: ");
    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++)
        for(j = 0; j < bno; j++)
            if(flags[j] == 0 && bsize[j] >= psize[i])
            {
                allocation[j] = i;
                flags[j] = 1;
                break;
            }
    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");
    }
}
```

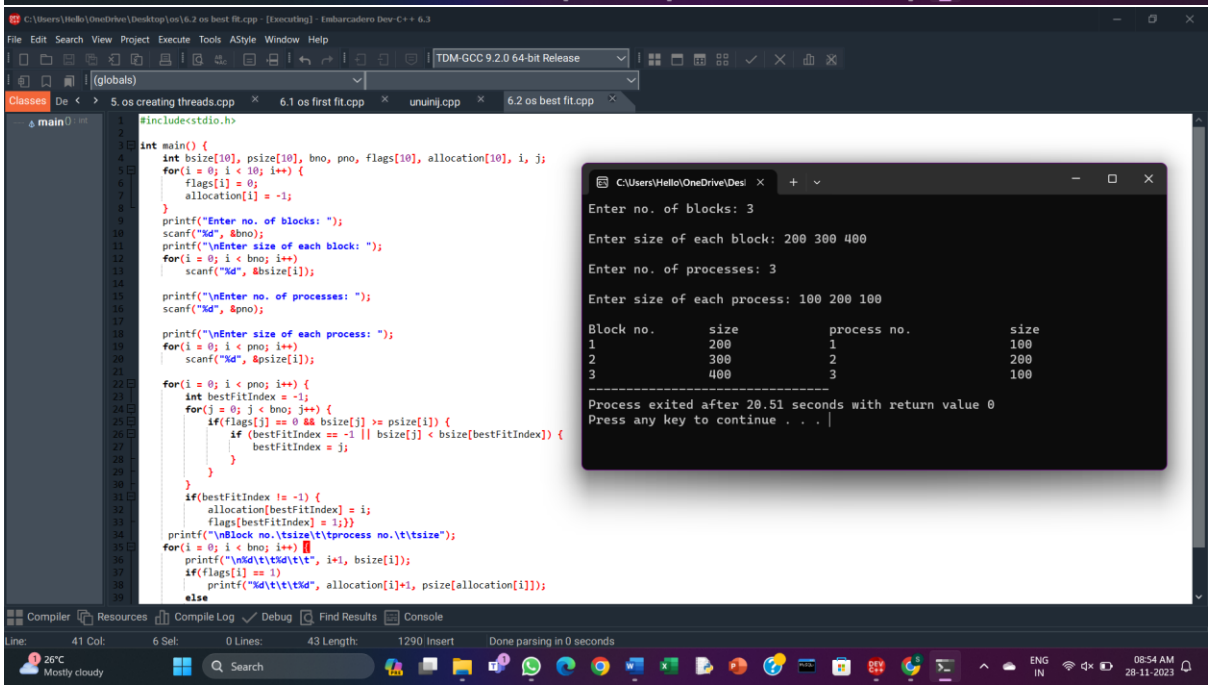
```
Enter no. of blocks: 3

Enter size of each block: 200 300 400

Enter no. of processes: 3

Enter size of each process: 100
200
300

Block no.       size            process no.             size
1               200             1                       100
2               300             2                       200
3               400             3                       300
--------------------------------
Process exited after 32.57 seconds with return value 0
Press any key to continue . . .
```



```
#include<stdio.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++) {
        int bestFitIndex = -1;
        for(j = 0; j < bno; j++) {
            if(flags[j] == 0 && bsize[j] >= psize[i]) {
                if (bestFitIndex == -1 || bsize[j] < bsize[bestFitIndex]) {
                    bestFitIndex = j;
                }
            }
        }
        if(bestFitIndex != -1) {
            allocation[bestFitIndex] = i;
            flags[bestFitIndex] = 1;}}
     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
```

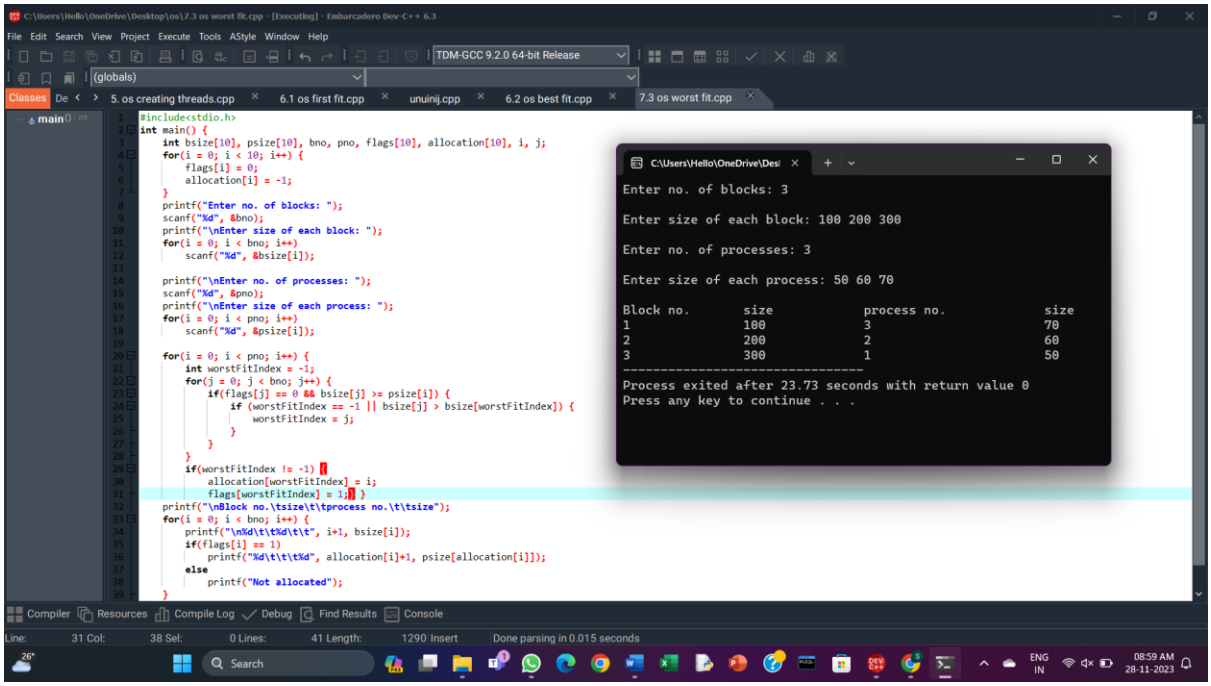
```
Enter no. of blocks: 3

Enter size of each block: 200 300 400

Enter no. of processes: 3

Enter size of each process: 100 200 100

Block no.       size            process no.             size
1               200             1                       100
2               300             2                       200
3               400             3                       100
--------------------------------
Process exited after 20.51 seconds with return value 0
Press any key to continue . . .
```


C:\Users\Hello\OneDrive\Desktop\os\7.3 os worst fit.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Classes
5. os creating threads.cpp
6.1 os first fit.cpp
unuinij.cpp
6.2 os best fit.cpp
7.3 os worst fit.cpp
main()
#include<stdio.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++) {
int worstFitIndex = -1;
for(j = 0; j < bno; j++) {
if(flags[j] == 0 && bsize[j] >= psize[i]) {
if (worstFitIndex == -1 || bsize[j] > bsize[worstFitIndex]) {
worstFitIndex = j;
}
}
}
if(worstFitIndex != -1) {
allocation[worstFitIndex] = i;
flags[worstFitIndex] = 1;
}
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");
}
C:\Users\Hello\OneDrive\Desi
Enter no. of blocks: 3
Enter size of each block: 100 200 300
Enter no. of processes: 3
Enter size of each process: 50 60 70
Block no. size process no. size
1 100 3 70
2 200 2 60
3 300 1 50
Process exited after 23.73 seconds with return value 0
Press any key to continue . . .
Compiler Resources Compile Log Debug Find Results Console
Line: 31 Col: 38 Sel: 0 Lines: 41 Length: 1290 Insert Done parsing in 0.015 seconds
Search
ENG IN
08:59 AM 28-11-2023