```cpp
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  struct Process
5  {
6      int id;
7      int size;
8      int allocation;
9      bool isGiven = false;
10 };
11
12 struct Memory
13 {
14     int size;
15     int free;
16     int allocated;
17     bool isTaken = false;
18     int extfrag;
19     int givenProcessId = -1;
20 };
21
22 int m;
23 int n;
24 int external_fragmentation = 0;
25 int internal_fragmentation = 0;
26
27
28
29 void worstFit(Process p[], int n, Memory mem[], int m)
30 {
31     int j = 0;
32     for (int i = 0; i < n; i++)
33     {
34         int worst_index = -1;
35         for (int j = 0; j < m; j++)
36         {
37             if (mem[j].size >= p[i].size)
38             {
39                 if (worst_index == -1)
40                     worst_index = j;
41                 else if (mem[worst_index].size < mem[j].size)
42                     worst_index = j;
43             }
44         }
45
46         if (worst_index != -1)
47         {
48             mem[worst_index].isTaken = true;
49             p[i].isGiven = true;
50             mem[worst_index].givenProcessId = p[i].id;
51             mem[worst_index].free -= p[i].size;
52             mem[worst_index].size -= p[i].size;
53             p[i].allocation = worst_index + 1;
54             mem[worst_index].allocated = p[i].id;
55
56         }
57     }
58
59
60 }
61
```

```cpp
62  void calcfrag(Process p[], int n, Memory mem[], int m)
63  {
64      int flag = 0;
65      for (int i = 0; i < m; i++)
66      {
67          if (mem[i].givenProcessId == -1)
68          {
69              flag = 1;
70              break;
71          }
72      }
73      if (flag == 0)
74      {
75          external_fragmentation = 0;
76      }
77      else
78      {
79          for (int i = 0; i < m; i++)
80          {
81              if (mem[i].isTaken != true || mem[i].givenProcessId == -1)
82              {
83                  external_fragmentation += mem[i].size;
84              }
85          }
86      }
87
88      for (int i = 0; i < m; i++)
89      {
90          if (mem[i].isTaken != false)
91          {
92              internal_fragmentation += mem[i].free;
93          }
94      }
95  }
96
97  void printTable(Process P[], int n, Memory mem[], int m, int memorySize[])
98  {
99
100     for (int i = 0; i < m; i++)
101     {
102         if (mem[i].free == memorySize[i])
103         {
104             mem[i].free = 0;
105         }
106     }
107
108     cout << "\nTable-->(-1 Denotes Unallocated process)\n";
109     int i;
110
111     puts("+-----+------------+-------------+----------------+");
112     puts("| BNO | Block Size | Process All. | Internal Fragm. |");
113     puts("+-----+------------+-------------+----------------+");
114
115     for (i = 0; i < m; i++)
116     {
117         printf("| %2d |     %2d     |      %2d      |       %3d       |\n", i + 1,
    memorySize[i], mem[i].givenProcessId, mem[i].free);
118         puts("+-----+------------+-------------+----------------+");
119     }
120
121     cout << "External Fragmentation: " << external_fragmentation << endl;
```

```cpp
122         cout << "Internal Fragmentation: " << internal_fragmentation << endl;
123 }
124
125 int main()
126 {
127     cout << "\nEnter the number of memory blocks: ";
128     cin >> m;
129     Memory mem[m];
130     int memorySize[m];
131     for (int i = 0; i < m; i++)
132     {
133         cout << "\n";
134         cout << "Enter the size of the memory block " << i + 1 << ": ";
135         cin >> mem[i].size;
136         mem[i].free = mem[i].size;
137         mem[i].allocated = -1;
138         mem[i].extfrag = 0;
139         memorySize[i] = mem[i].size;
140     }
141
142     cout << "\nEnter the number of processes: ";
143     cin >> n;
144     Process p[n];
145     for (int i = 0; i < n; i++)
146     {
147         p[i].id = i + 1;
148         cout << "\n";
149         cout << "\nEnter the size of the process" << p[i].id << ": ";
150         cin >> p[i].size;
151     }
152
153     worstFit(p, n, mem, m);
154
155     calcfrag(p, n, mem, m);
156
157     printTable(p, n, mem, m, memorySize);
158 }
```

Code - os lab - Visual Studio Code

File   Edit   Selection   View   Go   Run   Terminal   Help

> Code   ✕   Screenshot from 2022-03-24 16-36-01.png

```
anzal 🔲 anzal 🔲 ../4th sem/os lab/p12 🔲 cd "/home/anzal/Desktop/4th sem/os lab/p12/" && g++ worstfit.cpp -o worstfit && "/home/anzal/Desktop/4th sem/os lab/p12/"worstfit

Enter the number of memory blocks: 5

Enter the size of the memory block 1: 200

Enter the size of the memory block 2: 100

Enter the size of the memory block 3: 300

Enter the size of the memory block 4: 400

Enter the size of the memory block 5: 500

Enter the number of processes: 4

Enter the size of the process1: 450

Enter the size of the process2: 210

Enter the size of the process3: 210

Enter the size of the process4: 250

Table-->(-1 Denotes Unallocated process)
+-----+------------+--------------+----------------+
| BNO | Block Size | Process All. | Internal Fragm. |
+-----+------------+--------------+----------------+
|  1  |    200     |     -1       |       0        |
+-----+------------+--------------+----------------+
|  2  |    100     |     -1       |       0        |
+-----+------------+--------------+----------------+
|  3  |    300     |      3       |      90        |
+-----+------------+--------------+----------------+
|  4  |    400     |      2       |     190        |
+-----+------------+--------------+----------------+
|  5  |    500     |      1       |      50        |
+-----+------------+--------------+----------------+
External Fragmentation: 300
Internal Fragmentation: 330

anzal 🔲 anzal 🔲 ../4th sem/os lab/p12 🔲
```

Run Testcases   ⊗ 0 ⚠ 0   🔒 anzal1 🔒   Live Share   Go Live