Question 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.

Garv nanwani
19BCS049

Code :

```
#include<bits/stdc++.h>
using namespace std;

void Worst_Fit(int block_size[], int total_blocks, int process_size[], int
total_process) {
   int allocation[total_process];
   memset(allocation, -1, sizeof(allocation));
   for (int i = 0; i < total_process; i++) {
       int worstIdx = -1;
      for (int j = 0; j < total_blocks; j++) {
         if (block_size[j] >= process_size[i]) {
           if (worstIdx == -1)
               worstIdx = j;
            else if (block_size[worstIdx] < block_size[j])
               worstIdx = j;
         }
      }
      if (worstIdx != -1) {
         allocation[i] = worstIdx;
         block_size[worstIdx] -= process_size[i];
      }
   }
   cout << "\nProcess No.\tProcess Size\tBlock no.\n";
   for (int i = 0; i < total_process; i++) {
      cout << " " << i+1 << "\t\t" << process_size[i] << "\t\t";
      if (allocation[i] != -1)
         cout << allocation[i] + 1;
      else
         cout << "Not Allocated";
         cout << endl;
   }
}
int main() {
   int total_blocks,total_process;
    cout << "Enter Number of Memory Blocks\n";
    cin >> total_blocks;
    cout << "Enter Number of Process\n";
    cin >> total_process;

   int block_size[total_blocks], process_size[total_process];
   cout << "Enter values of Memory Blocks\n";
   for (int i = 0; i < total_blocks; i++) {
       cin >> block_size[i];
   }
   cout << "Enter values of Process\n";
   for (int i = 0; i < total_process; i++) {
       cin >> process_size[i];
   }
   Worst_Fit(block_size, total_blocks, process_size, total_process);
   return 0 ;
}
```

Output :

```
Enter Number of Memory Blocks
5
Enter Number of Process
5
Enter values of Memory Blocks
4
5
6
7
8
Enter values of Process
4
5
3
7
8

Process No.     Process Size    Block no.
 1              4               5
 2              5               4
 3              3               3
 4              7               Not Allocated
 5              8               Not Allocated

--------------------------------
```