

C++ implementation of worst - Fit algorithm

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
// Function to allocate memory to blocks as per worst fit
```

```
// algorithm
```

```
void worstFit(int blockSize[], int m, int processSize[],
```

```
int
```

```
n)
```

```
{
```

```
// Stores block id of the block allocated to a
```

```
// process
```

```
int allocation[n];
```

```
// Initially no block is assigned to any process
```

```
memset(allocation, -1, sizeof(allocation));
```

```
// pick each process and find suitable blocks
```

```
// according to its size and assign to it
```

```
for (int i=0; i<n; i++)
```

```
{
```

```
// Find the best fit block for current process
```

```
int wstIdx = -1;
```

```
for (int j=0; j<m; j++)
```

```
{
```

```
if (blockSize[j] >= processSize[i])
```

```
{
```

```
if (wstIdx == -1)
```

```

wstIdx = j;

else if (blockSize[wstIdx] < blockSize[j])

wstIdx = j;

}

}

// If we could find a block for current process

if (wstIdx != -1)

{

// allocate block j to p[i] process

allocation[i] = wstIdx;

// Reduce available memory in this block.

blockSize[wstIdx] -= processSize[i];

}

}

cout << "\nProcess No.\tProcess Size\tBlock no.\n";

for (int i = 0; i < n; i++)

{

cout << " " << i+1 << "\t\t" << processSize[i] << "\t\t";

if (allocation[i] != -1)

cout << allocation[i] + 1;

else

cout << "Not Allocated";

cout << endl;

}

}

// Driver code

```

```
int main()
{
int blockSize[] = {100, 500, 200, 300, 600};
int processSize[] = {212, 417, 112, 426};
int m = sizeof(blockSize)/sizeof(blockSize[0]);
int n = sizeof(processSize)/sizeof(processSize[0]);
worstFit(blockSize, m, processSize, n);
return 0 ;
}
```

OUTPUT:

Process No. Process Size Block no.

1 212 5

2 417 2

3 112 5

4 426 Not Allocated