

B.BHANUTEJA REDDY-192325016

22. Construct a C program to implement the best fit algorithm of memory management.

AIM:

To construct a C program to implement the Best Fit algorithm for memory management.

ALGORITHM:

1. Input the number of memory blocks and their sizes.
2. Input the number of processes and their sizes.
3. For each process, find the smallest memory block that can accommodate the process (best fit).
4. Allocate the process to the selected block and reduce the block size.
5. If no suitable block is found, mark the process as unallocated.
6. Display the allocation details.

PROCEDURE:

1. Input the number of memory blocks and their sizes.
2. Input the number of processes and their sizes.
3. Loop through each process to find the smallest block that can fit the process.
4. Allocate the process to the block and adjust the block's size.
5. Print the allocation results.

CODE:

```
#include <stdio.h>
```

```
int main() {
```

```
    int nb, np;
```

```
    scanf("%d", &nb);
```

```

int blockSize[nb];

for (int i = 0; i < nb; i++) {
    scanf("%d", &blockSize[i]);
}

scanf("%d", &np);

int processSize[np], allocation[np];

for (int i = 0; i < np; i++) {
    scanf("%d", &processSize[i]);
    allocation[i] = -1;
}

for (int i = 0; i < np; i++) {
    int bestIdx = -1;
    for (int j = 0; j < nb; j++) {
        if (blockSize[j] >= processSize[i]) {
            if (bestIdx == -1 || blockSize[j] < blockSize[bestIdx]) {
                bestIdx = j;
            }
        }
    }

    if (bestIdx != -1) {
        allocation[i] = bestIdx;
        blockSize[bestIdx] -= processSize[i];
    }
}

printf("\nProcess No.\tProcess Size\tBlock Allocated\n");

for (int i = 0; i < np; i++) {
    printf("%d\t\t%d\t\t", i + 1, processSize[i]);
    if (allocation[i] != -1)

```

```

        printf("%d\n", allocation[i] + 1);

    else

        printf("Not Allocated\n");

}

return 0;

}

```

OUTPUT:

The screenshot shows the OnlineGDB web interface. On the left is a sidebar with navigation links: 'Welcome, BandlapalliBhanutejareddy', 'Create New Project', 'My Projects', 'Classroom' (with a 'new' badge), 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main area displays the output of a C++ program. The output shows memory addresses for four processes: 5, 100, 300, 400, 200, 500, 4, 212, 515, 606, and 203. Below this, a table summarizes the process information:

Process No.	Process Size	Block Allocated
1	212	2
2	515	Not Allocated
3	606	Not Allocated
4	203	3

At the bottom, a green message states: '...Program finished with exit code 0' and 'Press ENTER to exit console.'