

Experiment:23-Construct a C program to implement the first fit algorithm of memory management.

Aim:

To implement the First Fit memory allocation algorithm to allocate processes to the first suitable memory block.

Procedure:

1. Input:

- Array of block sizes.
- Array of process sizes.

2. First Fit Allocation:

- For each process, search through the blocks sequentially to find the first block that can accommodate the process.
- Allocate the block to the process and reduce the block size by the size of the process.
- If no suitable block is found, mark the process as "Not Allocated".

3. Output:

- Display the allocation of blocks to processes, showing the process number, process size, and allocated block number. If a process is not allocated, display "Not Allocated".

Code

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

```
void firstFit(int blockSize[], int m, int processSize[], int n) {
```

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

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

```
        allocation[i] = -1;
```

```
    }
```

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

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

```

        if (blockSize[j] >= processSize[i]) {
            allocation[i] = j;
            blockSize[j] -= processSize[i];
            break;
        }
    }
}

```

```

printf("Process No.\tProcess Size\tBlock no.\n");
for (int i = 0; i < n; i++) {
    printf("%d\t\t%d\t\t", i + 1, processSize[i]);
    if (allocation[i] != -1) {
        printf("%d", allocation[i] + 1);
    } else {
        printf("Not Allocated");
    }
    printf("\n");
}
}

```

```

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]);

    firstFit(blockSize, m, processSize, n);
    return 0;
}

```

Output:

Output			
Process No.	Process Size	Block no.	
1	212	2	
2	417	5	
3	112	2	
4	426	Not Allocated	