

13. Construct a C program for implementation the various memory allocation strategies.

```
#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("\nProcess No.\tProcess Size\tBlock No.\n");
    for (int i = 0; i < n; i++) {
        printf("%d\t%d\t", i + 1, processSize[i]);
        if (allocation[i] != -1)
            printf("%d\n", allocation[i] + 1);
        else
            printf("Not Allocated\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]);  
printf("First Fit Allocation:\n");  
firstFit(blockSize, m, processSize, n);  
  
return 0;  
}
```

OUTPUT:

First Fit Allocation:

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