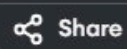


main.c



Run

Output

Clear

```
1 #include <stdio.h>
2 #define MAX 100
3 void firstFit(int blockSize[], int m, int processSize[], int n) {
4     int allocation[MAX];
5     for (int i = 0; i < MAX; i++)
6         allocation[i] = -1;
7     for (int i = 0; i < n; i++) {
8         for (int j = 0; j < m; j++) {
9             if (blockSize[j] >= processSize[i]) {
10                 allocation[j] = i;
11                 blockSize[j] -= processSize[i];
12                 break;
13             }
14         }
15     }
16     printf("Process No.\tProcess Size\tBlock no.\n");
17     for (int i = 0; i < n; i++) {
18         printf("%d\t\t%d\t\t", i + 1, processSize[i]);
19         if (allocation[i] != -1)
20             printf("%d\n", allocation[i] + 1);
21         else
22             printf("Not Allocated\n");
```

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

=== Code Execution Successful ===



Search



ENG
IN



8:07 PM
12/15/2024

main.c



Share

Run

Output

Clear

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <fcntl.h>
4 #include <unistd.h>
5 #include <sys/types.h>
6 #include <sys/stat.h>
7 int main() {
8     int fd;
9     char *filename = "example.txt";
10    char *text = "Hello, UNIX System Calls!\n";
11    fd = open(filename, O_CREAT | O_WRONLY | O_TRUNC, S_IRUSR |
12             S_IWUSR);
13    if (fd == -1) {
14        perror("Error opening file");
15        exit(EXIT_FAILURE);
16    }
17    if (write(fd, text, sizeof(text)) == -1) {
18        perror("Error writing to file");
19        close(fd);
20        exit(EXIT_FAILURE);
21    }
22    close(fd);
```

Error opening file: Permission denied

=== Code Exited With Errors ===