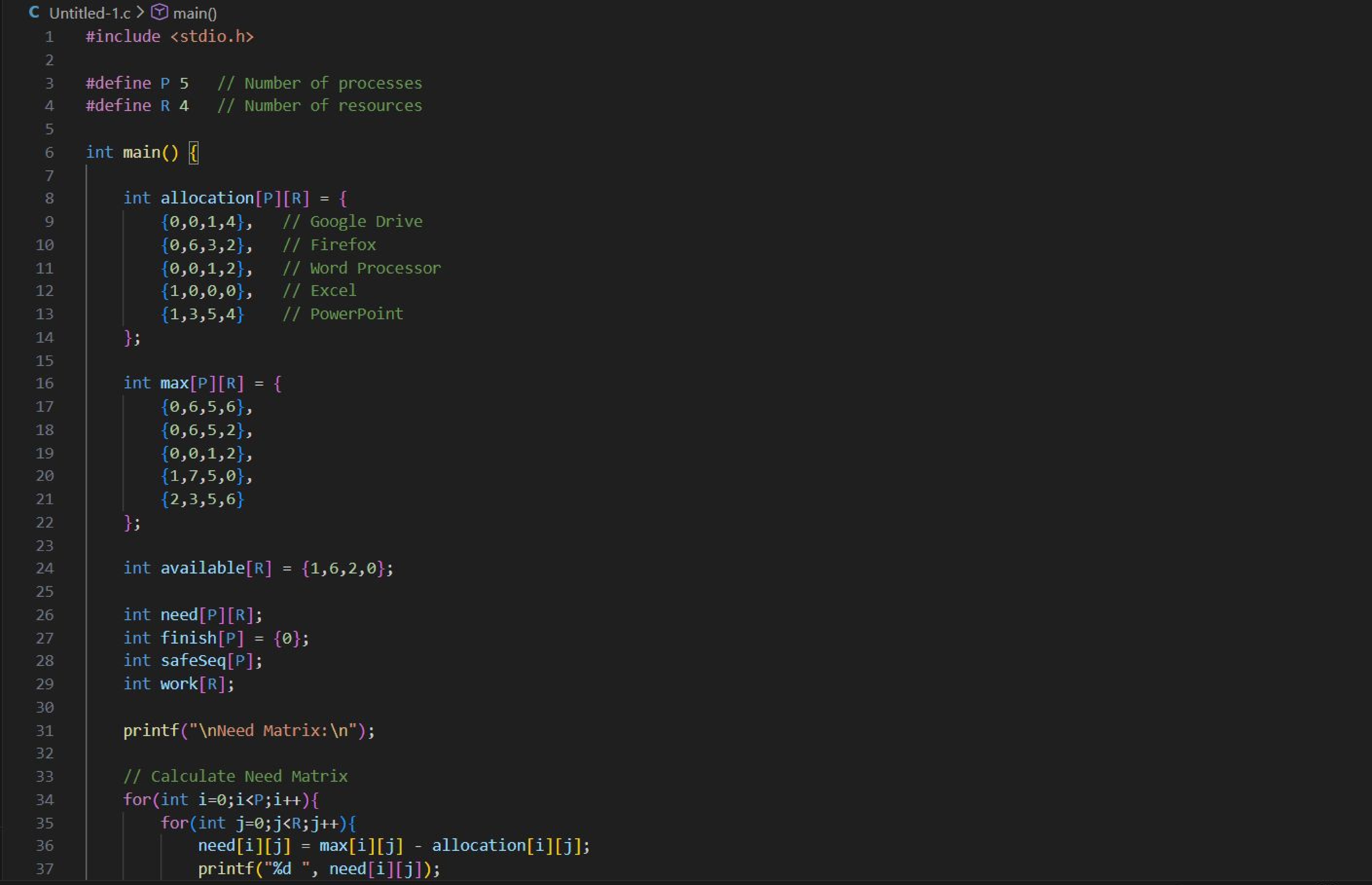
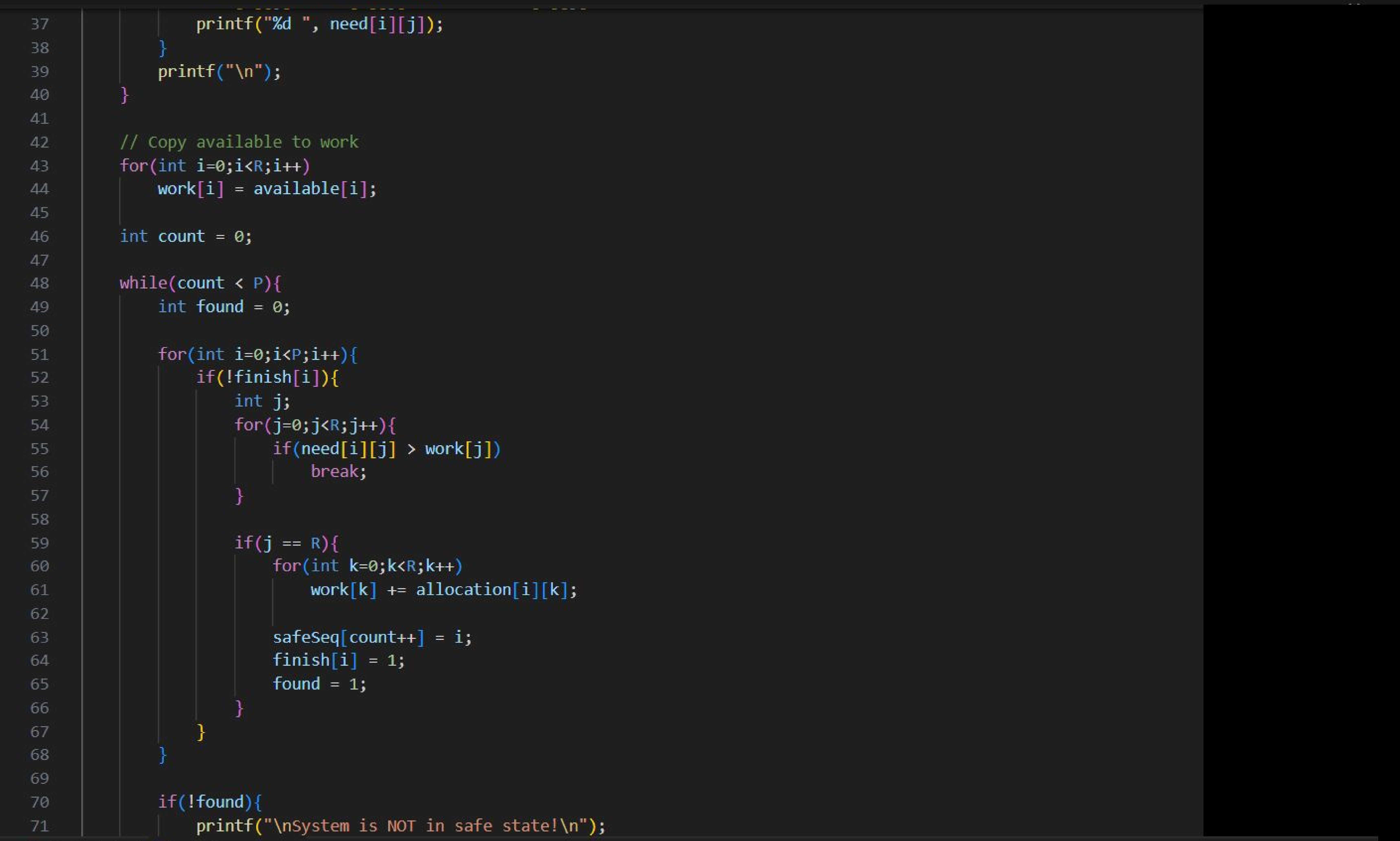
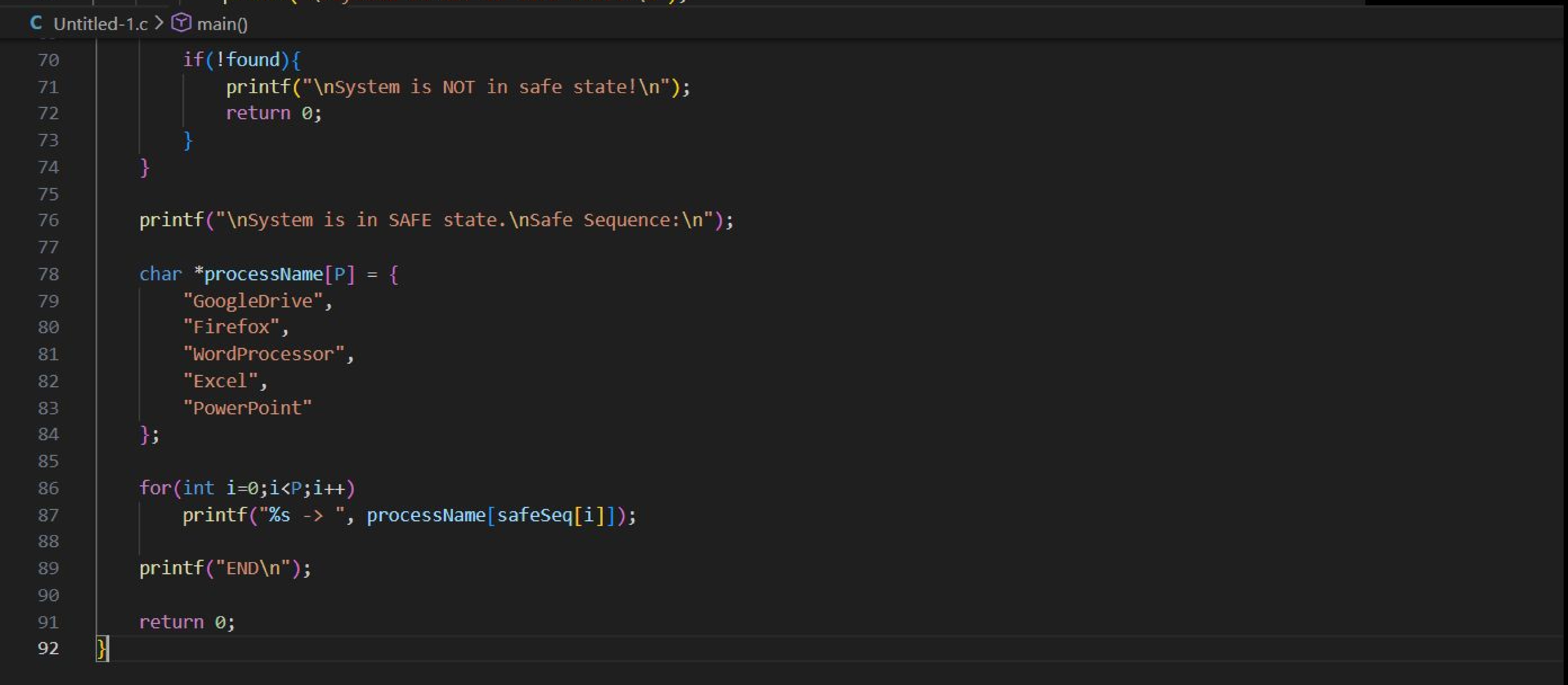


C Untitled-1.c > main()

```c
#include <stdio.h>

#define P 5   // Number of processes
#define R 4   // Number of resources

int main() {

    int allocation[P][R] = {
        {0,0,1,4},   // Google Drive
        {0,6,3,2},   // Firefox
        {0,0,1,2},   // Word Processor
        {1,0,0,0},   // Excel
        {1,3,5,4}    // PowerPoint
    };

    int max[P][R] = {
        {0,6,5,6},
        {0,6,5,2},
        {0,0,1,2},
        {1,7,5,0},
        {2,3,5,6}
    };

    int available[R] = {1,6,2,0};

    int need[P][R];
    int finish[P] = {0};
    int safeSeq[P];
    int work[R];

    printf("\nNeed Matrix:\n");

    // Calculate Need Matrix
    for(int i=0;i<P;i++){
        for(int j=0;j<R;j++){
            need[i][j] = max[i][j] - allocation[i][j];
            printf("%d ", need[i][j]);
```

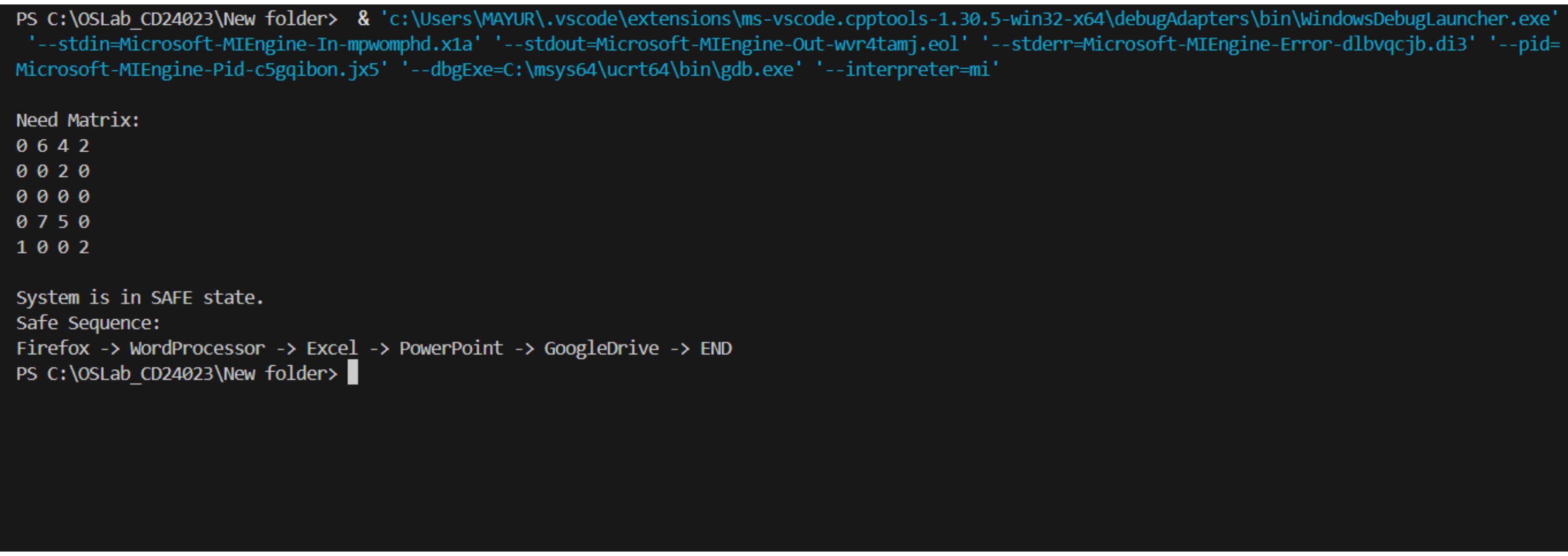


```c
            printf("%d ", need[i][j]);
        }
        printf("\n");
    }

    // Copy available to work
    for(int i=0;i<R;i++)
        work[i] = available[i];

    int count = 0;

    while(count < P){
        int found = 0;

        for(int i=0;i<P;i++){
            if(!finish[i]){
                int j;
                for(j=0;j<R;j++){
                    if(need[i][j] > work[j])
                        break;
                }

                if(j == R){
                    for(int k=0;k<R;k++)
                        work[k] += allocation[i][k];

                    safeSeq[count++] = i;
                    finish[i] = 1;
                    found = 1;
                }
            }
        }

        if(!found){
            printf("\nSystem is NOT in safe state!\n");
```



C Untitled-1.c > main()

```c
        if(!found){
            printf("\nSystem is NOT in safe state!\n");
            return 0;
        }
    }

    printf("\nSystem is in SAFE state.\nSafe Sequence:\n");

    char *processName[P] = {
        "GoogleDrive",
        "Firefox",
        "WordProcessor",
        "Excel",
        "PowerPoint"
    };

    for(int i=0;i<P;i++)
        printf("%s -> ", processName[safeSeq[i]]);

    printf("END\n");

    return 0;
}
```



```
PS C:\OSLab_CD24023\New folder>  & 'c:\Users\MAYUR\.vscode\extensions\ms-vscode.cpptools-1.30.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe'
 '--stdin=Microsoft-MIEngine-In-mpwomphd.x1a' '--stdout=Microsoft-MIEngine-Out-wvr4tamj.eol' '--stderr=Microsoft-MIEngine-Error-dlbvqcjb.di3' '--pid=
Microsoft-MIEngine-Pid-c5gqibon.jx5' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

Need Matrix:
0 6 4 2
0 0 2 0
0 0 0 0
0 7 5 0
1 0 0 2

System is in SAFE state.
Safe Sequence:
Firefox -> WordProcessor -> Excel -> PowerPoint -> GoogleDrive -> END
PS C:\OSLab_CD24023\New folder> 
```