

3 0 5

Примерен изход:

```
###(H H)[XXX](O O)(O O)#####  
#####  
##(H H)(O O)(O O)(O O)(O O)(O O)
```

Решение:

За решението на задачата ще използваме три едномерни масива, в които ще записваме съответно – в pos[i]- позицията на локомотива в i-тия коловоз, в c[i] – броя на вагоните с въглища, а в f[i]- броя на цистерните.

Единствената важна преценка е да се изчисли броят на символите, оставащи след последния вагон.

Те се изчисляват, като от дължината на коловоза М се извади:

$5 * (1 + F + C) + \text{позиция на локомотива}$, където F е броя на вагоните с въглища, а C- броя на цистерните.

Реализация на алгоритъма на езика C++

```
#include <cstdlib>  
#include <iostream>  
using namespace std;  
int main()  
{  
    int i,j,m,n;  
    int pos[9],c[9],f[9];  
    cin>>n>>m;  
    for (i=0;i<n;i++)  
        cin>>pos[i]>>c[i]>>f[i];  
    for (i=0;i<n;i++)  
    {  
        if (pos[i]>0)  
        {  
            for (j=1;j<pos[i];j++) cout<<'#';  
            cout<<"(H H)";  
            for (j=0;j<c[i];j++) cout<<"[XXX]";  
            for (j=0;j<f[i];j++) cout<<"(O O)";  
            for (j=5*(1+c[i]+f[i])+pos[i];j<=m;j++) cout<<'#';  
        }  
        else  
            for (j=0;j<m;j++) cout<<'#';  
        cout<<'\\n';  
    }  
}
```

Задача D3. Библиотека

Първата задача на библиотекарката Книжка Книгова на новото и работно място не била от най – любимите и. Тя трябвало да подреди N книги на шкаф с много рафтове, така че след като някой поиска дадена книга да може да я намира за най – малко време.

Ако подреди всичките N книги само на един рафт, времето необходимо за намирането, на коя да е книга ще е N секунди, а ако раздели книгите на K рафта, то времето, необходимо за намирането на книга от кой да е рафт ще е $K + P$ секунди, където P е броя на книгите на съответния рафт (т.е. K секунди, за да намери рафта и още P секунди, за да намери търсената книга). Например, ако има 7 книги, Книжка може да сложи всички книги на един рафт и тогава ще и отнеме 7 секунди за да намери коя да е от тях. Ако тя раздели 7 книги на 2 рафта с по 3 и 4 книги всеки, то ще и отнеме първо 2 секунди за да намери правилния рафт и след това още 3 или 4 секунди за намиране на всяка една от книгите, но след като незнаем коя книга ще търсим общото време за търсене ще бъде 6 секунди. Помогнете на библиотекарката да разбере колко ще е минималното време, необходимо намиране на произволна книга.

Напишете програма **MINITIME.EXE**, която от стандартния вход чете едно цяло число $N(1 \leq N \leq 107)$ – броя на книгите, които Книжка трябвало да подреди и отпечатва на стандартния изход едно число – минималното време, за което тя ще може да намери коя да е от книгите.

Примерен вход:

7

Примерен изход:

6

Примерен вход:

3

Примерен изход:

3

Решение:

Очевидно е, че трябва да намираме минималното време разпределяйки равномерно книгите по рафтове, започвайки от разпределение на книгите на един рафт, след това на два рафта и така разпределяйки ги до $N/2$ рафта включително. Тук интересно е смятането на времето за всяко едно разпределение по рафтове. Ако разделим N книги на K рафта то имаме две възможности. Първата е ако книгите се разпределят по равно на всичките K рафта т.е. K дели N без остатък. Тогава минималното време ще е K секунди (броя на рафтовете) и още N/K секунди – броя на книгите на един рафт. Но ако книгите не могат да се разделят по равно на всичките K рафта, тогава трябва да ги разпределим на $K+1$ рафта, където първите K рафта ще са с по равен брой книги, а $K+1$ -вия ще е с толкова книги колкото са останали след разполагането на първите K рафта т.е. с $N \% K$ книги. Тогава минималното време ще е $K+1$ секунди (за броя на рафтовете) и още N/K секунди за броя на книгите, защото на $K+1$ -вия рафт броя на книгите ще е винаги по – малък от колкото в горните рафтове.

Реализация на алгоритъма на езика C++

```
#include <iostream>
using namespace std;
int main()
{
    int n,mint,i,p,t;
    cin>>n;
    mint=n;
```