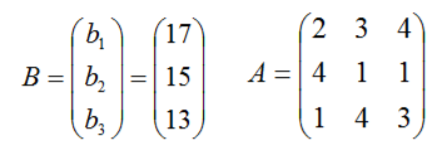
# Компьютерное моделирование на основе решения систем линейных уравнений

## 1 задача

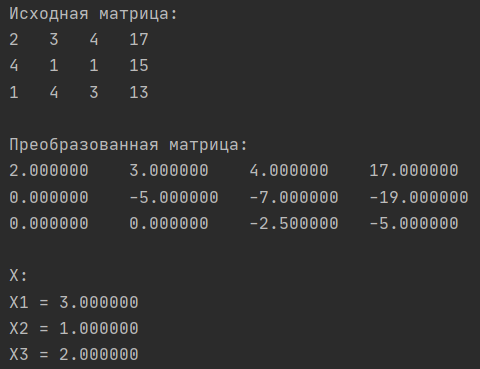
Из порта необходимо вывезти 3 вида груза в количестве b1, b2, b3. В распоряжении имеется 3 типа судов, причем j-е судно может взять на борт aij единиц i-го груза. Определить количество судов каждого типа x1, x2, x3, которое необходимо привлечь для вывоза груза при следующих исходных данных:



### Код программы

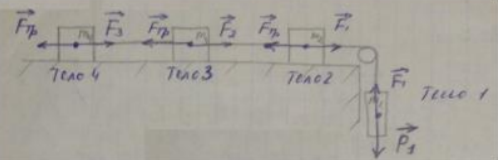
#include <stdio.h>  
#define n 3  
  
int main()  
{  
 int i, j, k;  
 float c, s;  
 float a[n][n+1] = {2, 3, 4, 17, 4, 1, 1, 15, 1, 4, 3, 13};  
 float x[n];  
  
 printf("Исходная матрица:\n");  
 for (i = 0; i < n; i++)  
 {  
 for (j = 0; j < n + 1; j++)  
 printf("%.0f\t", a[i][j]);  
 printf("\n");  
 }  
  
 for (i = 0; i < n - 1; i++)  
 {  
 for (k = i + 1; k < n; k++)  
 if (a[k][i] != 0)  
 {  
 {  
 c = a[k][i] / a[i][i];  
 for (j = i; j < n + 1; j++)  
 a[k][j] = a[k][j] - c \* a[i][j];  
 }  
 }  
 }  
  
 printf("\nПреобразованная матрица:\n");  
 for (i = 0; i < n; i++)  
 {  
 for (j = 0; j < n + 1; j++)  
 printf("%f\t", a[i][j]);  
 printf("\n");  
 }  
  
 x[n-1] = a[n-1][n] / a[n-1][n-1];  
 for (i = 2; i >= 0; i--)  
 {  
 s = 0;  
 for (j = i + 1; j < n; j++)  
 s += a[i][j] \* x[j];  
 x[i] = (a[i][n] - s) / a[i][i];  
 }  
  
 printf("\nX:\n");  
 for (i = 0; i < n; i++)  
 printf("X%d = %f\n", i + 1, x[i]);  
 return 0;  
}

### Результат



## 2 задача

Имеется система 4-х грузов, связанных нитью (рис 1). Три груза расположены на горизонтальной плоскости. Система приводится в движение грузом, который движется в вертикальном направлении и подвешен на нити, перекинутой через неподвижный блок.



Рисунок

Определить ускорение движения системы и силы натяжения каждой нити, если m1 = m2 = m3 = m4 = 1 кг. Коэффициент трения kтр = 0,01, ускорение свободного падения g = 9,8 м/с^2.

### Код программы

#include <stdio.h>  
#define n 4  
  
int main() {  
 int i, j, k;  
 float c, s;  
 float a[n][n+1] = {-1, 0, 0, -1, -9.8, 1, -1, 0, -1, 0.098, 0, 1, -1, -1, 0.098, 0, 0, 1, -1, 0.098};  
 float x[n];  
  
 printf("\nИсходная матрица:\n");  
 for (i = 0; i < n; i++)  
 {  
 for (j = 0; j < n + 1; j++)  
 printf("%.3f\t", a[i][j]);  
 printf("\n");  
 }  
  
 for (i = 0; i < n - 1; i++)  
 {  
 for (k = i + 1; k < n; k++)  
 if (a[k][i] != 0)  
 {  
 {  
 c = a[k][i] / a[i][i];  
 for (j = i; j < n + 1; j++)  
 a[k][j] = a[k][j] - c \* a[i][j];  
 }  
 }  
 }  
  
 printf("\nПреобразованная матрица:\n");  
 for (i = 0; i < n; i++)  
 {  
 for (j = 0; j < n+1; j++)  
 printf("%f\t", a[i][j]);  
 printf("\n");  
 }  
  
 x[n-1] = a[n-1][n] / a[n-1][n-1];  
 for (i = 2; i >= 0; i--)  
 {  
 s = 0;  
 for (j = i + 1; j < n; j++)  
 s += a[i][j] \* x[j];  
 x[i] = (a[i][n] - s) / a[i][i];  
 }  
  
 printf("\nX:\n");  
 for (i = 0; i < n - 1; i++)  
 printf("F%d = %.4f\n", i + 1, x[i]);  
 printf("a = %.5f\n", x[3]);  
 return 0;  
}

### Результат

