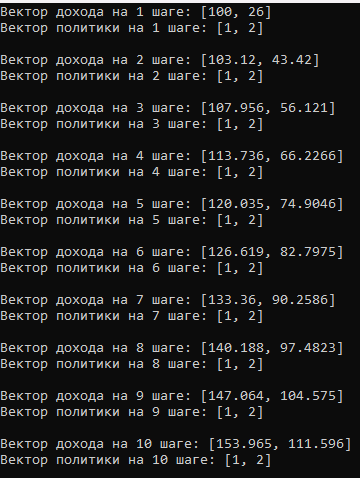
Задание 1.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | #include <iostream>  #include <vector>  #include <algorithm>  **using** **namespace** std;  **int** **main**()  {  setlocale(LC\_ALL, "Russian");  **int** T = **10**;  **int** r1 = **12**, r2 = **17**, c1 = **5**, c2 = **7**;  **double** p1 = **0.12**, q1 = **0.23**, p2 = **0.32**, q2 = **0.33**;  **double** p1o = **1** - p1;  **double** p2o = **1** - p2;  **double** q1o = **1** - q1;  **double** q2o = **1** - q2;  vector<**double**> v = { **100**, **0** };  vector<**double**> f = { **0**, **0** };  vector<**double**> kd = { **0**, **0**};  vector<**double**> kr = { **0**, **0**};  **for** (**int** i = **1**; i < T+**1**; i++) {  kd[**0**] = r1 + p1o \* v[**0**] + p1 \* v[**1**];  kd[**1**] = r2 + p2o \* v[**0**] + p2 \* v[**1**];  kr[**0**] = -c1 + q1 \* v[**0**] + q1o \* v[**1**];  kr[**1**] = -c2 + q2 \* v[**0**] + q2o \* v[**1**];  **auto** max\_kd = max\_element(kd.begin(), kd.end());  **auto** max\_kr = max\_element(kr.begin(), kr.end());  v[**0**] = \*max\_kd;  v[**1**] = \*max\_kr;  cout << "Вектор дохода на " << i << " шаге: [" << v[**0**] << ", " << v[**1**] << "]" << endl;  **for** (**int** k = **0**; k < kd.size(); ++k) {  **if** (kd[k] == v[**0**]) {  f[**0**] = k;  **break**;  }  }  **for** (**int** m = **0**; m < kr.size(); ++m) {  **if** (kr[m] == v[**1**]) {  f[**1**] = m;  **break**;  }  }  cout << "Вектор политики на " << i << " шаге: [" << f[**0**]+**1** << ", " << f[**1**]+**1** << "]" << endl;  cout << endl;  }  **return** **0**;  } |



Задание 2:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57 | B = **0.6**  r = [**12**, **17**];  c = [**5**, **7**];  p = [**0.12**, **0.23**];  q = [**0.32**, **0.33**];  po = [**1**-p(**1**), **1** - p(**2**)];  qo = [**1**-q(**1**), **1** - q(**2**)];  f = [**1**, **1**];  kd = [**0**, **0**];  kr = [**0**, **0**];  w = cell(**1**, **0**);  i = **1**;  **while** (true)  disp(sprintf('Step %d:', i));  A = [**1**-po(f(**1**))\*B, -p(f(**1**))\*B; -B\*q(f(**2**)), **1**-qo(f(**2**))\*B];  b = [r(f(**1**)); -c(f(**2**))];  % Решение СЛУ  x = A \ b;  % Вывод результата  disp('Solution x: ');  disp(x);  w{**end**+**1**} = x;    disp('Strat norm: ');  kd(**1**) = r(**1**) + po(**1**)\*B\*w{i}(**1**) + p(**1**)\*B\*w{i}(**2**)  kr(**1**) = -c(**1**) + q(**1**)\*B\*w{i}(**1**) + qo(**1**)\*B\*w{i}(**2**)      disp('Strat strength: ');  kd(**2**) = r(**2**) + po(**2**)\*B\*w{i}(**1**) + p(**2**)\*B\*w{i}(**2**)  kr(**2**) = -c(**2**) + q(**2**)\*B\*w{i}(**1**) + qo(**2**)\*B\*w{i}(**2**)    % Находим максимум в векторе  max\_value = max(kd);  % Находим индекс (номер столбца) максимального элемента  index\_of\_max = find(kd == max\_value);    % Находим максимум в векторе  max\_value1 = max(kr);  % Находим индекс (номер столбца) максимального элемента  index\_of\_max1 = find(kr == max\_value1);    f1 = [index\_of\_max, index\_of\_max1]    are\_equal = (f == f1);    **if** are\_equal  **break**;  **else**  f = f1;  i++;  **end**  **end** |

