1. Банкерын алгоритмыг ашиглан дараах системийн төлөв дээр deadlock үүссэн эсэхийг шалгах програм бичнэ.

```
#include <stdio.h>
int main()
{
      int n, m, i, j, k;
      n = 5; // Number of processes
      m = 4; // Number of resources
      int alloc[5][4] = { { 0, 0, 1, 2 }, // P0
                          { 1, 0, 0, 0 }, // P1
                          { 1, 3, 5, 4 }, // P2
                          { 0, 6, 3, 2 }, // P3
                          { 0, 0, 1, 4 } }; // P4

      int max[5][4] = { { 0, 0, 1, 2 }, // P0
                        { 1, 7, 5, 0 }, // P1
                        { 2, 3, 5, 6 }, // P2
                        { 0, 6, 5, 2 }, // P3
                        { 0, 6, 5, 6 } }; // P4

      int avail[4] = { 1, 5, 2, 0 };

      int f[n], ans[n], ind = 0;
      for (k = 0; k < n; k++) {
            f[k] = 0;
      }
      int need[n][m];
      for (i = 0; i < n; i++) {
            for (j = 0; j < m; j++)
                  need[i][j] = max[i][j] - alloc[i][j];
      }
      int needs[m];
      int y = 0;
      for (k = 0; k < 5; k++) {
            for (i = 0; i < n; i++) {
                  if (f[i] == 0) {

                        int flag = 0;
                        for (j = 0; j < m; j++) {
                              if (need[i][j] > avail[j]){
                                    flag = 1;
                                    break;
                              }
                        }

                        if (flag == 0) {
                              ans[ind++] = i;
                              for (y = 0; y < m; y++){
                                    avail[y] += alloc[i][y];
                                    needs[y] += need[i][y];
                              }
                              f[i] = 1;
                        }
                  }
            }
      }
      int cnt = 0;
      for(i = 0; i < m; i++){
            if(needs[i] == avail[i]) cnt++;
      }
```

```
      if(cnt == m) printf("Yes. Deadlock was created.\n");
      else printf("No, Deadlock wasn't created.)
      return 0;
}
```

2. Process болгоны хувьд Need матрицыг харуулна.
P0-ээс P4 хүртэлх процессуудын Need - ийн утгууд нь Max(0,0,1,2) - allocation(0,0,1,2) = (0,
0, 0, 0), (0, 7, 5, 0), (1, 0, 0, 2), (0, 0, 2, 0), (0, 6, 4, 2).

3. P1 process Need(0,4,2,0) хүсэлттэй бол энэ хүсэлтийг шууд биелүүлэх боломжтой юу?
Available - ийн утга (1, 1, 0, 0) - ээс эхлэх үед энэ хүсэлтийг шууд биелүүлэх боломжтой. Дуусгах боломжтой процессуудын нэг дараалал нь P0, P2, P3, P1, P4 юм.