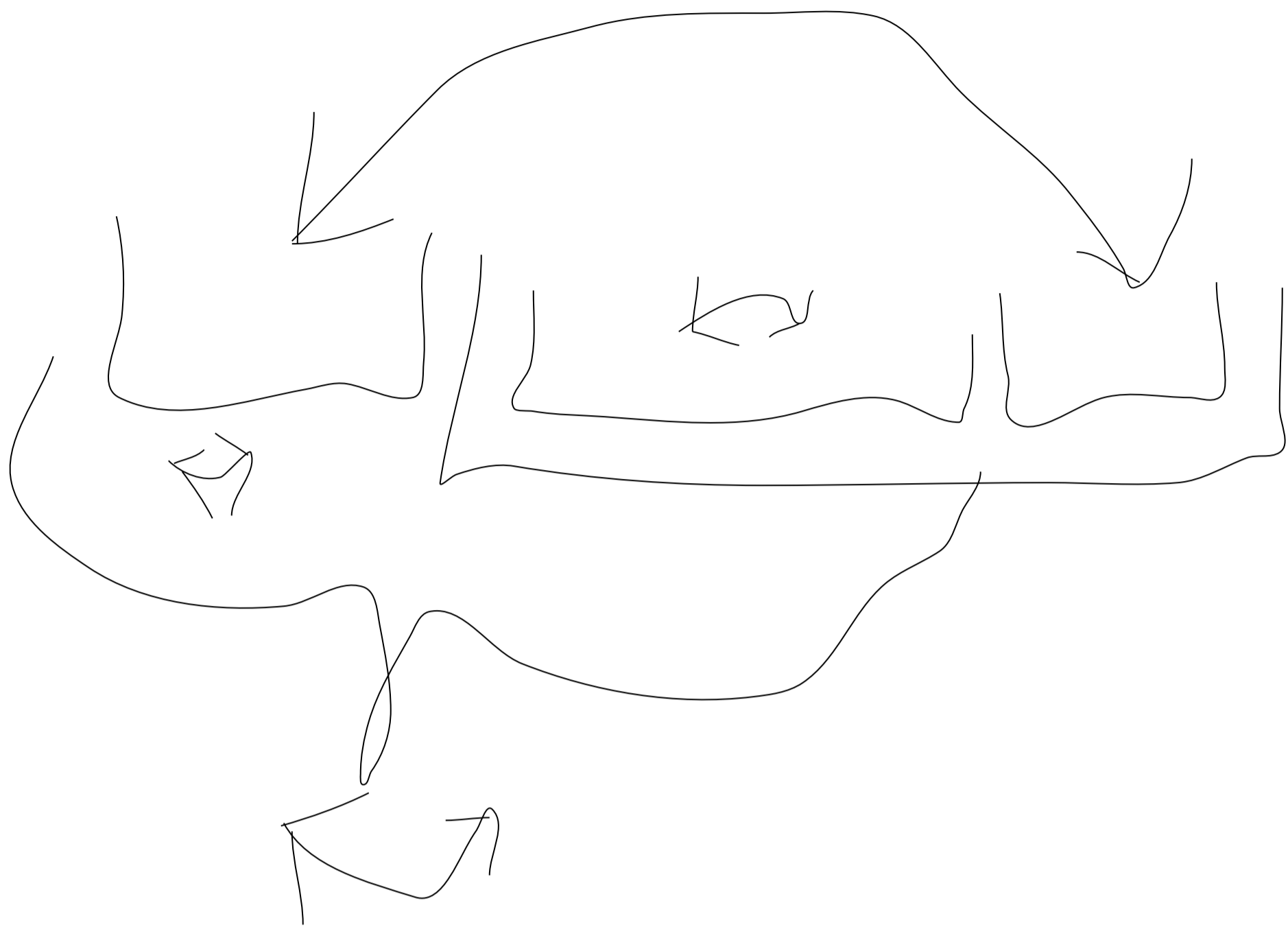


(применение суперреверса) переставить какой-то начальный кусок массива с оставшимся 1 5 6 9 2 => 6 9 2 1 5

d=2



1 2 3 4 5 6 7 8 9
 7 8 9 4 5 6 1 2 3
 9 8 7 6 5 4 1 2 3
 4 5 6 7 8 9 1 2 3

1 2 3 4 5 6 7 8 9
 3 2 1 4 5 6 7 8 9
 3 2 1 9 8 7 6 5 4
 4 5 6 7 8 9 1 2 3

```
void megareverse(int ms[], int razmer, int space)
{
    int t1;
    int k1=0;
    while(k1<(space-1)/2)
    {
        t1=ms[k1];
        ms[k1]=ms[space-1-k1];
        ms[space-1-k1]=t1;
        k1++;
    }
    int t2;
    int k2=space;
    while(k2<(razmer-1-space)/2+space)
    {
        t2=ms[k2];
        ms[k2]=ms[razmer-1-k2+space];
        ms[razmer-1-k2+space]=t2;
        k2++;
    }
    int t3;
    int k3=0;
    while(k3<(razmer-1)/2)
    {
        t3=ms[k3];
        ms[k3]=ms[razmer-1-k3];
        ms[razmer-1-k3]=t3;
        k3++;
    }
}
```

```
void megareverse(int ms[], int razmer, int space)
{
    int d, kri;
    kri=razmer-space;
    if(kri>space)
    {
        int i=0;
        while(i<space)
        {
            d=ms[i];
            ms[i]=ms[kri+i];
            ms[kri+i]=d;
            i++;
        }
        int t1;
        int k1=0;
        while(k1<=(space-1)/2)
        {
            t1=ms[k1];
            ms[k1]=ms[space-1-k1];
            ms[space-1-k1]=t1;
            k1++;
        }
        int t2;
        int k2=space;
        while(k2<=(kri-1-space)/2+space)
        {
            t2=ms[k2];
            ms[k2]=ms[kri-k2-1+space];
            ms[kri-1-k2+space]=t2;
            k2++;
        }
        int t3;
        int k3=0;
        while(k3<=(kri-1)/2)
        {
            t3=ms[k3];
            ms[k3]=ms[kri-1-k3];
            ms[kri-1-k3]=t3;
            k3++;
        }
    }
    else if(kri==space)
    {
        int i=0;
        while(i<space)
        {
            d=ms[i];
            ms[i]=ms[kri+i];
            ms[kri+i]=d;
            i++;
        }
    }
    else
    {
        int i=0;
        while(i<kri)
        {
            d=ms[i];
            ms[i]=ms[space+i];
            ms[space+i]=d;
            i++;
        }
        int t1;
        int k1=kri;
        while(k1<=(space-1-kri)/2+kri)
        {
            t1=ms[k1];
            ms[k1]=ms[space-1-k1+kri];
            ms[space-1-k1+kri]=t1;
            k1++;
        }
        int t2;
        int k2=space;
        while(k2<=(razmer-1-space)/2+space)
        {
            t2=ms[k2];
            ms[k2]=ms[razmer-1-k2+space];
            ms[razmer-1-k2+space]=t2;
            k2++;
        }
        int t3;
        int k3=kri;
        while(k3<=(razmer-1-kri)/2+kri)
        {
            t3=ms[k3];
            ms[k3]=ms[razmer-1-k3+kri];
            ms[razmer-1-k3+kri]=t3;
            k3++;
        }
    }
}
```

```
void superreverse(int ms[], int razmer, int start, int finish)
{
    int d;
    int i=start;
    while(i<(finish-start)/2+start)
    {
        d=ms[i];
        ms[i]=ms[finish-i+start];
        ms[finish-i+start]=d;
        i++;
    }
}

void megareverse(int ms[], int razmer, int space)
{
    superreverse(ms,razmer,0,space-1);
    superreverse(ms,razmer,space,razmer-1);
    superreverse(ms,razmer,0,razmer-1);
}
```