Поле игры жизнь - 2-у мерный динамический массив
1)нарастить измерение на восток/запад/юг/север
2)убавить измерение на восток/запад/юг/север

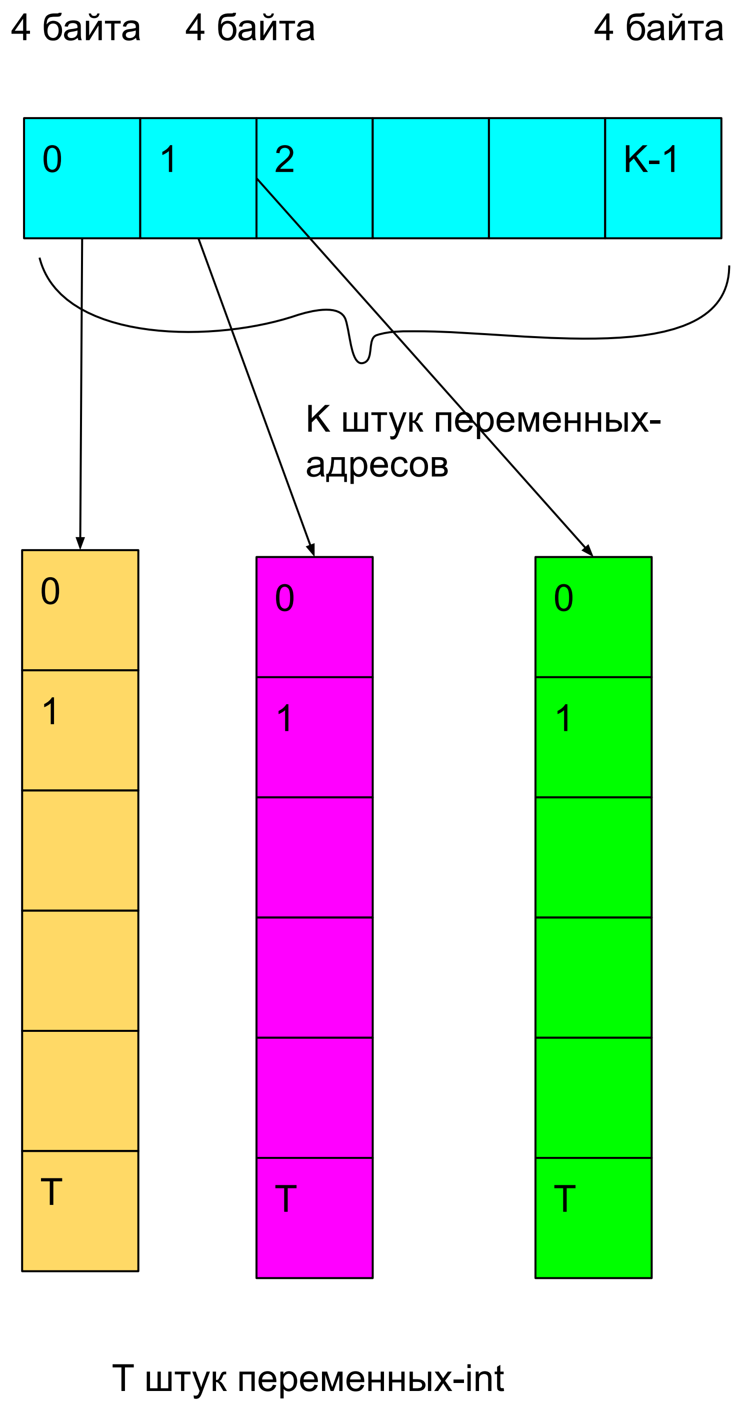
pandas
nymphy

R
Python

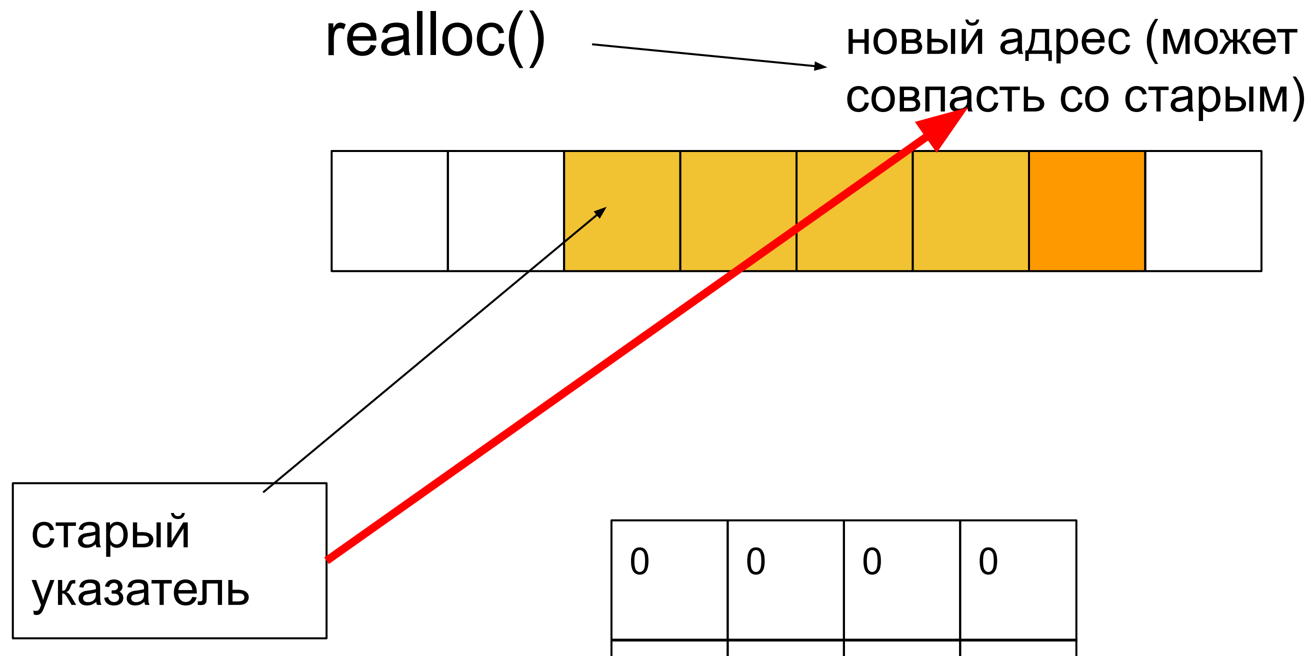
нейросети

СОЗДАНИЕ 2-у мерных
динамических массивов

1 способ
```
int **M=new int*[K];
for(int i=0;i<T;i++)
{
    M[i]=new int[T];
}
```

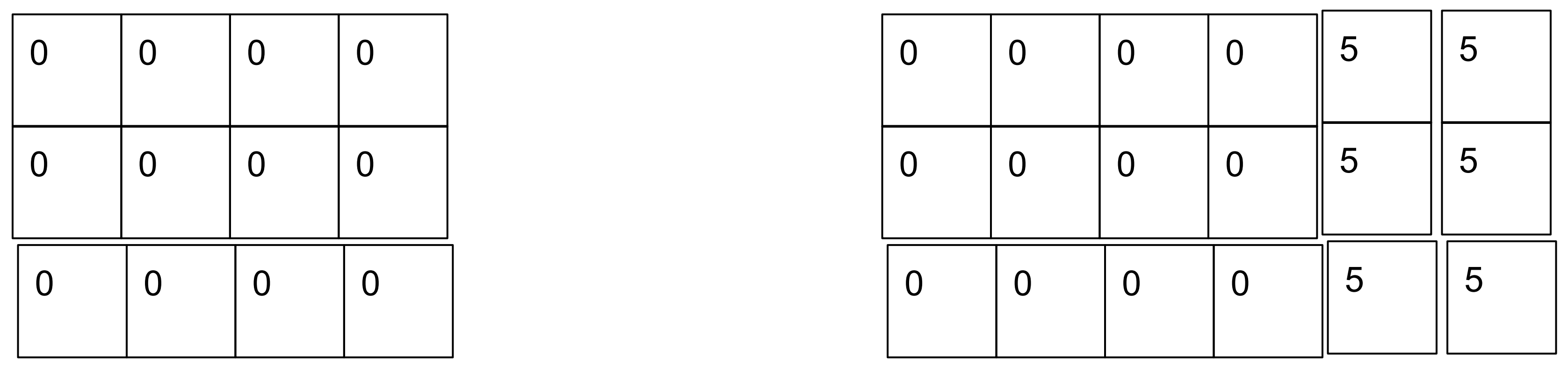


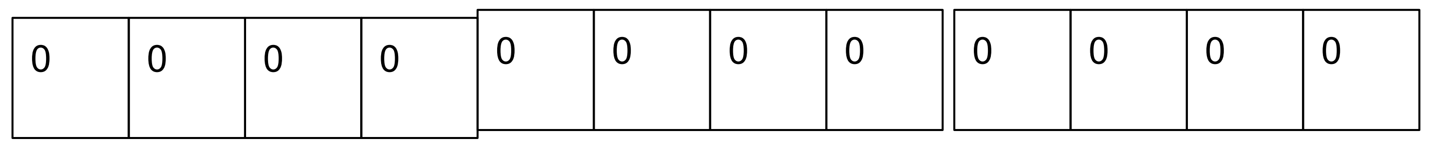
2 способ
```
int **M=new int*[K];
int *r=new int[K*T];
for(int i=0;i<T;i++)
{
    M[i]=r+i*T;
}
```

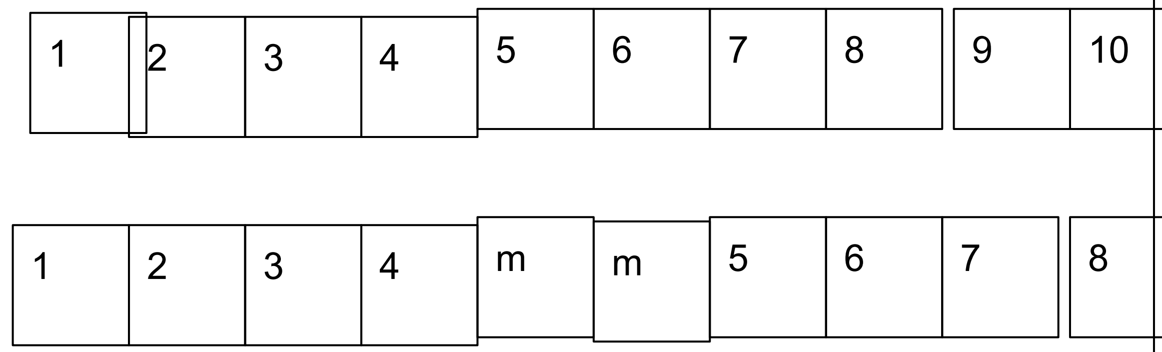

realloc()

новый адрес (может совпасть со старым)

старый указатель





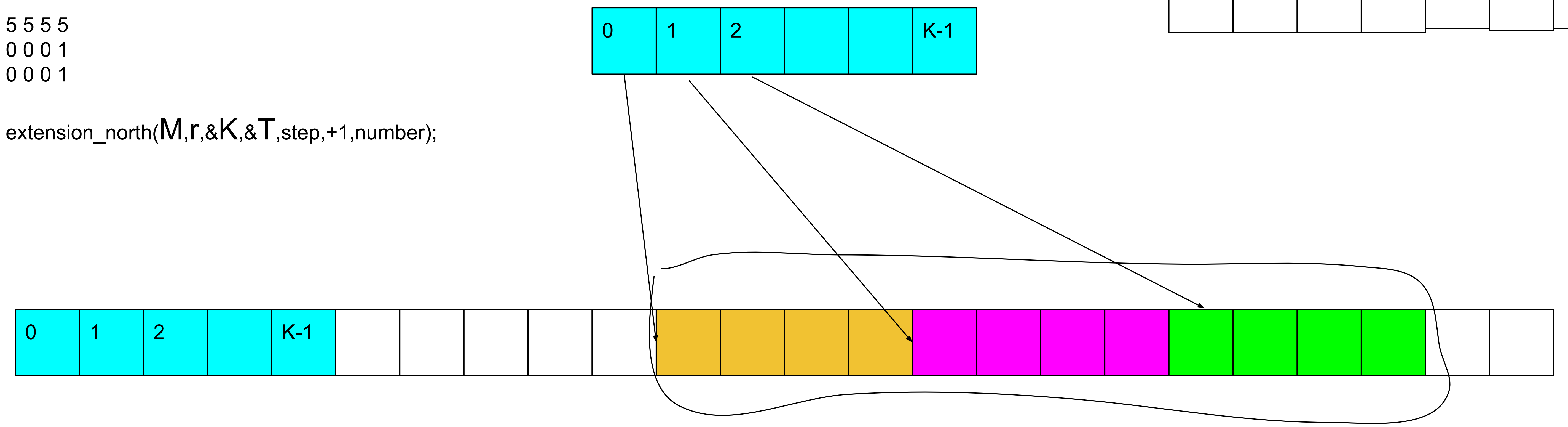


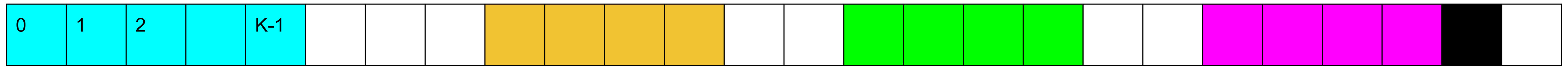
0 0 0
0 0 0

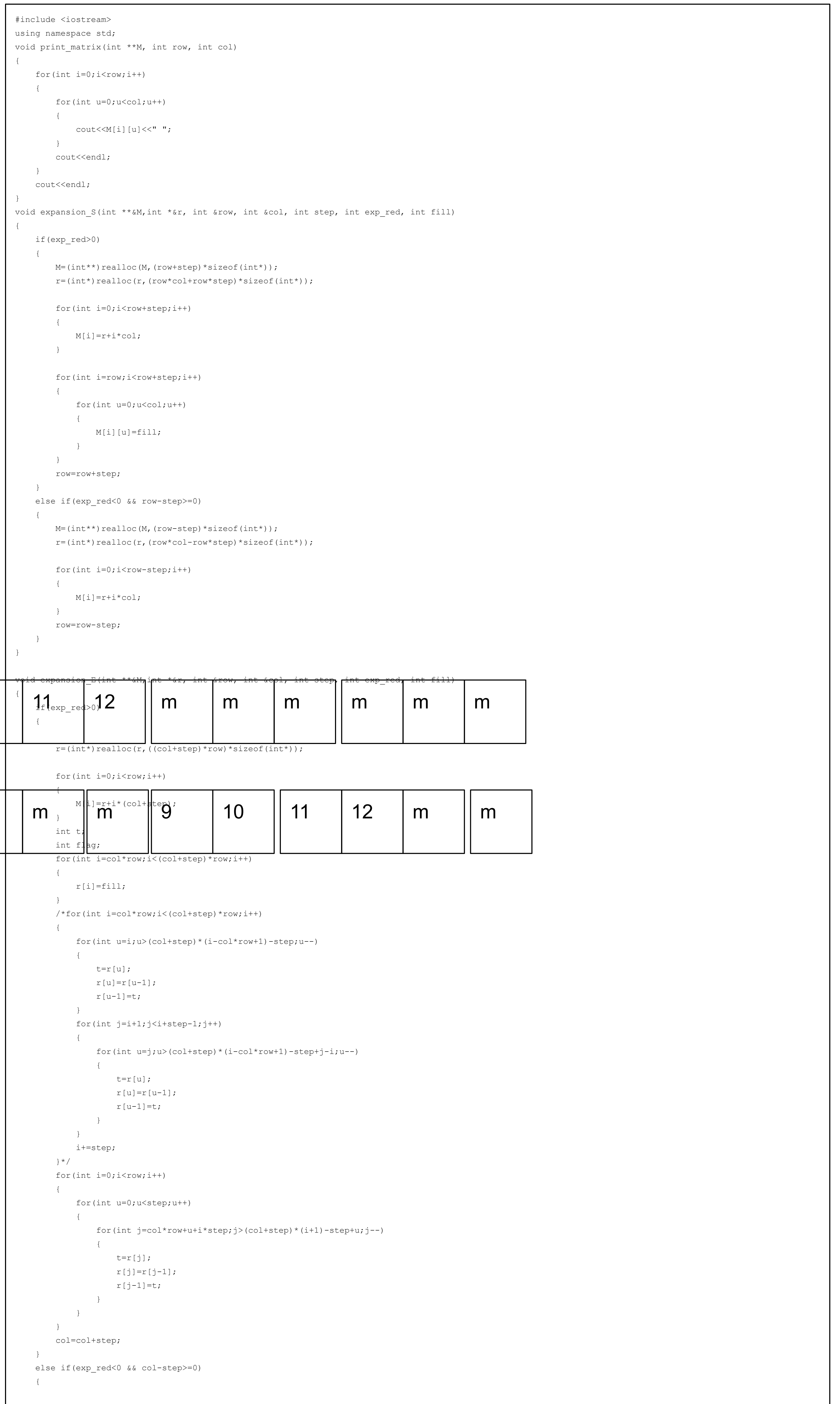
0 0 0 1
0 0 0 1

5 5 5 5
0 0 0 1
0 0 0 1

extension_north(M,r,&K,&T,step,+1,number);






```
#include <iostream>
using namespace std;
void print_matrix(int **M, int row, int col)
{
    for(int i=0;i<row;i++)
    {
        for(int u=0;u<col;u++)
        {
            cout<<M[i][u]<<" ";
        }
        cout<<endl;
    }
    cout<<endl;
}
void expansion_S(int **&M,int *&r, int &row, int &col, int step, int exp_red, int fill)
{
    if(exp_red>0)
    {
        M=(int**)realloc(M,(row+step)*sizeof(int*));
        r=(int*)realloc(r,(row*col+row*step)*sizeof(int*));

        for(int i=0;i<row+step;i++)
        {
            M[i]=r+i*col;
        }

        for(int i=row;i<row+step;i++)
        {
            for(int u=0;u<col;u++)
            {
                M[i][u]=fill;
            }
        }
        row=row+step;
    }
    else if(exp_red<0 && row-step>=0)
    {
        M=(int**)realloc(M,(row-step)*sizeof(int*));
        r=(int*)realloc(r,(row*col-row*step)*sizeof(int*));

        for(int i=0;i<row-step;i++)
        {
            M[i]=r+i*col;
        }
        row=row-step;
    }
}

void expansion_E(int **&M,int *&r, int &row, int &col, int step, int exp_red, int fill)
{
    if(exp_red>0)
    {

        r=(int*)realloc(r,((col+step)*row)*sizeof(int*));

        for(int i=0;i<row;i++)
        {
            M[i]=r+i*(col+step);
        }
        int t;
        int flag;
        for(int i=col*row;i<(col+step)*row;i++)
        {
            r[i]=fill;
        }
        /*for(int i=col*row;i<(col+step)*row;i++)
        {
            for(int u=i;u>(col+step)*(i-col*row+1)-step;u--)
            {
                t=r[u];
                r[u]=r[u-1];
                r[u-1]=t;
            }
            for(int j=i+1;j<i+step-1;j++)
            {
                for(int u=j;u>(col+step)*(i-col*row+1)-step+j-i;u--)
                {
                    t=r[u];
                    r[u]=r[u-1];
                    r[u-1]=t;
                }
            }
            i+=step;
        }*/
        for(int i=0;i<row;i++)
        {
            for(int u=0;u<step;u++)
            {
                for(int j=col*row+u+i*step;j>(col+step)*(i+1)-step+u;j--)
                {
                    t=r[j];
                    r[j]=r[j-1];
                    r[j-1]=t;
                }
            }
        }
        col=col+step;
    }
    else if(exp_red<0 && col-step>=0)
    {
```