

1) Programmanalyse

Zeile	i	j	e	m_ptr	n_ptr
6	8	6	{3, 2, 5, 7, 4}	o	o
7	8	6	{3, 2, 5, 7, 4}	j	o
8	8	6	{3, 2, 5, 7, 4}	j	i
9	5	6	{3, 2, 5, 7, 4}	j	i
10	5	6	{3, 6, 5, 7, 4}	j	i
11	5	5	{3, 6, 5, 7, 4}	j	i
12	5	5	{3, 6, 5, 7, 4}	j	e[3]
13	5	5	{3, 6, 5, 3, 4}	j	e[3]
14	5	2	{3, 6, 5, 3, 4}	j	e[3]
16	5	2	{3, 6, 5, 3, 7}	j	e[4]
17	5	2	{3, 6, 5, 3, 7}	e[3]	e[4]
18	5	2	{7, 6, 5, 3, 7}	e[3]	e[4]
19	5	2	{7, 6, 5, 3, 3}	e[4]	e[4]
20	5	2	{7, 6, 5, 3, 2}	e[4]	e[4]

2, 3) n-dimensionale Vektoren, dynamische mehrdim. Arrays

```
#include <iostream>

void add(const float *a, const float *b, float *c, int n)
{
    for(int i = 0; i < n; ++i)
        c[i] = a[i] + b[i];
}

float dot(const float *a, const float *b, int n)
{
    float d = 0;
    for(int i = 0; i < n; ++i)
        d += a[i] * b[i];
    return d;
}

void print(const float *a, int n)
{
    std::cout << "[" << a[0];
    for(int i = 1; i < n; ++i)
        std::cout << ", " << a[i];
    std::cout << "]\n";
}
```

```

int main()
{
    int n;
    std::cout << "n = ";
    std::cin >> n;

    // vektoren allozieren
    float *a = new float[n];
    float *b = new float[n];
    float *c = new float[n];

    // vektoren initialisieren
    for(int i = 0; i < n; ++i)
    {
        a[i] = 1 + i;
        b[i] = 1 + i + n;
    }

    // c = a + b
    add(a, b, c, n);

    // ausgabe
    std::cout << "a   = "; print(a, n);
    std::cout << "b   = "; print(b, n);
    std::cout << "a+b = "; print(c, n);

    // skalar produkt
    std::cout << "a.b = " << dot(a, b, n) << std::endl;

    // vektoren deallozieren
    delete[] a, b, c;

    // matrix allozieren
    float **m = new float*[n];
    for(int i = 0; i < n; ++i)
        m[i] = new float[n];

    // matrix initialisieren
    for(int i = 0; i < n; ++i)
        for(int j = 0; j < n; ++j)
            m[i][j] = 1 + j + i * n;

    // ausgabe
    std::cout << "M   = ";
    for(int i = 0; i < n; ++i)
    {
        print(m[i], n);
        std::cout << "      ";
    }

    // matrix deallozieren
    for(int i = 0; i < n; ++i)
        delete[] m[i];
    delete[] m;

    return 0;
}

```