

# 浙江工业大学

## 面向对象程序设计

2022/2023 (2)



### 实验一      复习

学生姓名      陈王子

学生学号      202103150503

学生班级      大数据分析 1 班

任课教师      毛国红

提交日期      2023 年 3 月 15 日

理学院

## 实验报告

### 第一题代码

```
#include <iostream>
#include <cstdlib>
#include <algorithm>
using namespace std;

int* InitArray(unsigned n) {
    int num;
    int* arr=new int[n];
    srand(time(NULL));
    for (int i = 0; i < n; i++) {
        do {
            num = rand() % 10000;
        } while (find(arr, arr + i, num) != arr + i);
        arr[i] = num;
    }
    cout << "Initialized" << endl;
    return arr;
}

void Display(int* p,unsigned n) {
    cout << "Displaying" << endl;
    for (int i = 0; i < n; i++) {
        cout << p[i] << " ";
    }
    cout << "Displayed" << endl;
}

int digitReverse(int num) {
    int reversedNum = 0;
    int pow10 = 100;
    while (num != 0) {
        int digit = num % 10;
        reversedNum += digit * pow10;
        pow10 /= 10;
        num /= 10;
    }
    return reversedNum;
}
```

```
void SortArray(int* p, unsigned n) {
    cout << "Sorting" << endl;
    for (int i = 0; i < n; i++) {
        int num = p[i];
        num = digitReverse(num);
        p[i] = num;
    }
    sort(p, p + n);
    cout << "Sorted" << endl;
}

void DestroyArray(int* arr) {
    delete arr;
    cout << "Deleted" << endl;
}

int main()
{
    unsigned n;
    int* p;
    while (cin >> n)
    {
        p = InitArray(n);
        //初始化一个规模为 n 的一维整型数组，并给出随机互异的值进行初始
        化。
        Display(p, n); //显示数据的初始化情况
        SortArray(p, n); //根据每个数包含的数字按个位、十位和百位和组
        成的新数值进行升序整理；如 123 对应的新数值为 321，12 的对应的新数值为
        210
        Display(p, n); //显示排序后的结果
        DestroyArray(p); //根据需要使用
    }
}
```

# 第一题测试

```
12
Initialized
Displaying
229 8478 2083 8328 2749 4462 6986 5034 8430 1129 9780 6515 Displayed
Sorting
Sorted
Displaying
34 87 264 380 430 515 689 823 874 921 922 947 Displayed
Deleted
123
Initialized
Displaying
236 9975 5043 919 3379 2316 3211 9179 8685 1600 9484 6604 3335 5303 5713 8610 1934 7856 4056 9020 1133 4753 1485 8636 2664 281 1992 88
1 1383 3042 938 2374 2935 8294 5924 4425 2525 9396 5081 9935 5773 3986 709 8454 1866 4597 8365 3275 6607 399 2905 8844 4045 8536 3088
1297 278 2368 9026 391 7657 8020 8732 124 5135 1295 440 3362 5750 371 3491 9166 9307 9720 883 144 638 9428 1800 5736 6384 2139 8481 16
87 9824 8409 7891 388 950 2686 4230 4032 7936 1338 7681 2492 5228 4761 9431 5097 4676 3024 4977 8224 3424 5209 6830 1715 6649 6987 816
8 499 410 2142 89 1220 9036 861 3550 4526 4258 4540 7579 Displayed
Sorting
Sorted
Displaying
6 8 14 16 20 20 22 27 32 38 44 45 55 57 59 112 134 167 168 173 180 182 184 186 188 193 194 198 230 237 240 241 263 294 299 303 317 331
340 357 377 383 388 406 420 421 422 424 428 429 439 441 448 454 466 473 483 484 492 509 517 524 525 531 533 539 539 540 563 572 579 5
84 586 592 613 620 625 630 632 635 636 637 639 650 658 661 668 676 686 699 693 703 706 756 779 786 789 790 792 795 822 824 833 836 839
852 861 863 872 880 883 902 904 907 919 931 946 971 973 975 980 993 994 Displayed
Deleted
```

```
Deleted
1234
Initialized
Displaying
239 755 139 4982 925 3474 3234 6252 5197 5452 9336 6648 2706 1849 7996 3180 5904 6281 5886 9776 2548 7466 3732 5651 7267 1340 533 7357 8608 3427 3939 743 6866 1256 3654 3101 4581 309 9660 8962 4944 919 3880 901 711 8677 1295 2643 4513
201 823 7084 1661 7788 1505 2507 6672 4934 1456 1904 7858 9686 3994 864 6539 229 5143 7216 9416 8247 4072 9757 7746 9746 7225 9552 2897 3153 7322 1280 9340 1611 746 8137 5572 6064 53 1595 1858 5891 641 5078 6542 1086 7235 69 722 9779
4791 8808 5342 1393 1135 1478 509 9412 8187 2466 3400 6599 6378 7182 1601 8790 6240 8301 9763 9582 2311 5070 720 1484 1694 9678 8083 1797 6804 4559 1582 6349 5435 6176 166 2378 2196 3316 9760 6845 50 3652 141 7112 4290 2816 2371 8865 4
562 7080 2245 313 4834 9042 2063 48 480 4043 4073 0416 7046 7983 2050 5203 4723 7092 3120 303 5771 5712 2112 1561 6221 5340 1064 9044 1353 3643 8852 8912 6787 268 9246 6182 528 2330 7037 9245 5645 4195 9782 7606 1691 918 461 2866 466 4
289 2074 7728 926 138 4179 2957 796 6093 7457 9904 3476 8642 2189 74049 2924 6707 3676 578 578 2207 7279 9380 5463 2366 133 8744 8586 5715 7559 1441 7413 4213 576 2253 4745 2683 599 120 9536 8308 9524 7115 2482 630 4482 3011 4196 6262 5973
8436 1400 9767 8527 7981 7480 9382 4590 557 4566 2263 2374 7938 7271 3049 6897 212 6429 1170 4023 3950 2322 5181 2753 3461 6893 537 3869 1251 7919 2709 1554 5124 3122 6328 881 995 9176 9363 6307 4750 2598 6447 4483 3185 4846 6665 1223
4058 6296 4774 52 2520 5218 8542 9378 937 1482 4826 553 5751 1344 6265 0850 8894 8748 1069 8171 225 3260 1228 2789 2318 3638 3863 1962 3295 2310 8432 4441 7857 1577 9102 4441 0882 2743 6862 965 9514 5514 4914 776 447 7926 868
8 5595 9646 1156 7713 6538 6697 7853 7076 3658 1322 5221 2640 3560 7444 3666 4175 8822 2720 6187 7648 172 9834 1925 9610 4858 8111 6584 9647 3881 4531 3980 370 7167 8834 561 7099 445 8117 2022 3073 4625 070 7112 4560 4226 406 7707 130
7 2697 5517 8579 4975 9393 3404 9442 8739 3349 7931 254 7550 5988 2711 847 2960 4029 8074 7860 6384 1363 5370 5588 6760 2532 4860 242 9231 4397 273 961 5355 686 1033 8229 3401 7877 8794 7585 4098 67 3795 339 6297 1153 8863 5592 1015 8
010 8717 8314 1443 9319 2122 6375 9043 9095 2686 545 1680 8859 9349 4085 6663 6458 8867 8974 856 9422 4571 8037 2917 637 510 8459 8723 9394 6933 4071 8364 1963 9635 3889 3226 7604 2535 6998 1409 6774 6966 3662 208 190 456 3275 4184 737
5 4386 2064 6699 9082 5600 4548 6646 1101 9172 2578 680 7753 5911 6680 5090 1225 9118 1521 3683 9232 4120 1014 1074 9626 6938 7937 895 6406 5083 7855 5455 5534 6304 9903 3204 6920 9047 887 608 8720 643 7827 101 5636 3331 112 2626 5081
324 3650 4143 3561 3582 72 0982 385 3127 8260 5837 9237 5255 4047 438 1080 672 118 2639 5760 8992 6040 8745 6525 2288 4488 5444 6997 2044 5110 8259 457 2584 9084 5713 2380 3804 3533 7097 9389 6294 1058 973 5365 5872 146 7035 5383 2492
7074 4879 1498 1715 9478 2122 4927 1684 8216 9154 5887 1039 2074 2729 2347 4422 4755 6898 4069 7548 2834 5596 6442 1204 6399 9862 391 9760 2036 702 4685 8716 8552 4043 5533 3603 2705 1832 2428 1877 1325 1919 1676 6351 4406 2638 1837 10
36 2746 3327 1190 8172 1579 3722 8820 9717 1271 5649 8146 3143 9115 6628 1370 2652 2484 3450 9995 2688 342 6083 1596 9461 5656 1381 9591 3962 1872 4617 117 2436 1970 1778 1072 4865 5122 2632 8903 4662 9894 985 329 9139 4504 928 698 486
3782 5039 4809 935 8077 5079 1915 7225 3115 2132 2700 5839 3623 5380 6131 3304 1417 4604 8537 5813 6344 4068 3145 5041 1616 7204 2237 4771 1760 4808 5334 8956 8474 1330 9879 2106 7230 1197 6173 3281 1467 6788 8166 7044 6691 3169 830 2
649 9218 4252 4849 2757 8239 5646 7848 6645 1132 3099 1509 8362 1345 6317 3303 332 7985 2616 8589 7391 5165 7593 1614 733 2675 449 11 782 8373 5943 3532 1743 6818 932 6475 9775 1257 7691 448 7109 1725 7377 253 1988 927 8522 7443 8597 2
491 9633 543 836 1639 8134 7618 1267 9089 2609 8662 2476 2235 718 2148 9733 7883 2426 4308 1026 4102 9980 2229 1705 5695 667 9088 2578 6986 4246 8712 9870 5830 3333 7253 570 5898 4015 2242 5860 6669 7581 2527 863 401 703 6380 8158 2509
5680 5931 5873 2768 6899 4096 4095 2217 4278 7124 5291 5534 513 4243 94049 451 8472 5211 8122 8731 3611 2090 544 2209 1255 1485 695 5677 7837 3457 3402 8523 8990 7545 2063 346 4867 3220 8704 687 731 4411 9673 7515 1229 6693 6813 4943
7704 8765 6702 4783 1653 5205 3152 4122 1463 3723 502 2639 8889 7021 5487 3741 6166 446 1361 8555 2882 6278 6386 9849 6360 2823 646 7395 7135 6886 3263 3852 1630 562 1290 4477 9472 1054 7920 9399 704 8691 5693 2343 1134 5244 2487 4785
5847 3238 1686 9963 9770 9061 9072 1242 2125 8461 7252 9162 9306 7617 677 1714 1379 1463 4277 7455 6544 399 3024 9706 1666 4312 326 8946 552 1258 1747 387 6745 5819 1768 4100 1094 3081 6152 4687 3758 7854 6935 3882 7346 7139 4906 7841
4736 5626 5834 2926 2553 6390 1700 1241 7870 432 86 4797 3433 4318 7491 2531 5496 235 777 4272 8780 65 9411 2490 290 2534 3298 7763 2739 3166 8491 4461 4584 4430 7856 3022 3619 4251 6587 7238 4420 4890 8311 64 2661 7040 1586 9161 6398
3042 1163 6062 2332 5086 5289 4063 4062 1574 2832 512 8314 200 9254 7227 9545 2380 3267 1835 8696 702 1600 1449 5855 4706 5445 4897 658 4079 4065 9059 8510 5168 5896 4533 3368 623 6598 7549 2019 2112 9392 2660 1609 4634 6590 8147 1161
2594 1326 1285 811 5270 0441 5162 9531 3169 8296 4042 292 2408 9255 8557 1420 8395 649 3393 7074 8046 7768 2240 7820 721 3897 6157 1866 7731 4394 15 870 7282 7895 5930 2459 4888 2335 4737 8566 8123 3642 6485 7897 9456 4680 741 5894 481
1691 6246 7588 6852 1081 5298 5079 9979 7543 3755 2 8324 1108 5864 8516 2478 7756 8789 1404 3843 2480 1319 5261 8996 6949 7789 9519 6044 4762 4697 2041 8141 4305 7147 379 5769 9509 1084 6767 2756 3684 3282 6090 4249 7295 7
727 9685 930 2032 5413 5223 1025 5377 4832 3903 5861 9090 5612 3757 2399 5367 1771 2781 1673 3818 3798 6640 8081 9156 5523 3262 8619 5573 1647 9972 6569 2369 9076 1770 6041 147 3380 5302 4548 5593 392 8032 3982 6227 4694 42 6758 758 92
48 2695 9560 269 2108 4534 1838 3347 3194 1387 181 7684 2197 106 4896 8927 5711 1626 2146 6047 5903 4235 0481 918 9548 Displayed
Sorting
Sorted
Displaying
1 2 3 4 5 7 7 10 11 13 15 16 19 21 21 22 24 26 27 28 28 29 32 33 34 36 36 38 38 39 39 40 42 43 43 43 44 46 46 46 50 50 50 52 54 55 56 57 57 59 62 62 63 65 65 66 67 67 67 67 67 68 68 69 69 70 70 71 71 72 72 73 73 73 74 75 77 77 78
78 80 81 82 82 83 83 83 84 84 85 86 86 86 86 87 89 90 90 90 91 91 92 92 92 93 94 95 95 97 98 100 101 101 101 102 102 103 104 104 109 110 110 111 112 113 114 114 116 117 117 117 118 120 122 122 125 125 127 132 133 135 136 137 137 13
7 137 138 139 139 140 141 141 142 144 144 144 146 146 147 147 148 152 152 153 154 154 156 156 157 159 160 160 161 161 162 163 164 164 164 164 165 165 166 166 169 169 170 171 172 172 173 174 174 175 177 177 177 180 180 180 181 181 1
81 182 183 184 184 185 185 187 188 188 188 188 192 193 193 194 194 194 196 196 196 197 198 201 201 201 203 206 206 206 207 208 210 211 212 213 213 215 216 216 216 217 217 218 220 220 220 221 221 221 221
223 223 224 224 225 227 228 230 230 231 231 231 232 232 233 233 234 234 235 235 235 237 238 238 238 239 240 240 242 242 242 243 243 244 244 245 245 246 246 247 248 250 250 251 251 251 252 252 254 254 255 255 255 258 258 261 261
262 262 262 263 265 265 266 266 267 268 268 269 269 269 270 270 270 271 271 271 272 272 274 274 275 276 276 278 278 279 280 280 281 282 282 283 284 285 285 285 287 287 287 288 289 289 289 292 293 293 294 295 298 300 30
0 302 303 303 383 306 306 309 309 309 309 309 310 312 313 313 314 314 315 315 317 317 318 320 320 321 322 322 325 325 326 326 327 327 328 328 330 331 331 334 335 335 335 335 336 337 337 339 339 340 340 341 341 341 342 343 3
43 344 344 344 345 345 346 346 346 347 347 347 349 349 350 351 351 352 352 352 353 353 355 357 357 358 360 361 362 362 363 364 364 367 367 368 368 368 369 369 369 370 371 372 373 373 376 376 376 378 378 379 379 379 380 384 386 386
388 388 389 390 393 393 395 395 396 396 398 402 402 403 404 404 405 406 406 407 408 409 410 410 410 413 415 415 415 416 417 419 420 421 421 423 423 425 429 431 433 433 435 435 435 436 438 438 438 439 440 443 443 444 444 445 445 447 449
449 450 451 452 452 452 455 455 456 458 460 460 463 466 467 468 468 469 469 469 470 470 470 470 473 473 473 474 474 476 477 477 479 479 480 480 480 481 482 483 484 484 484 485 485 485 485 486 486 486 487 489 490 490 491 493 493 495 495 496 496 49
7 498 498 499 503 503 504 505 505 507 507 510 510 510 511 511 515 517 517 517 519 520 521 522 522 525 526 527 527 529 529 529 530 530 530 531 531 532 532 532 533 534 535 536 537 539 542 542 543 544 545 545 546 547 547 548 5
50 552 552 552 553 554 554 556 557 557 557 558 559 560 560 560 561 563 568 569 571 571 572 573 574 574 576 576 577 578 580 581 582 583 585 586 586 587 589 589 589 590 590 591 592 592 592 593 593 595 595 596 597 598 598 599
599 601 601 603 604 604 604 606 606 607 607 609 609 609 610 611 612 612 613 614 614 615 616 616 617 618 620 622 622 623 624 624 625 626 626 626 626 629 629 629 630 630 633 634 634 636 637 638 639 640 641 641 641 642 643 643 644 646
646 647 647 647 648 649 649 651 651 651 654 654 654 656 657 657 658 658 660 661 661 661 663 663 664 664 664 665 665 666 666 669 669 670 671 674 674 675 676 677 677 677 679 680 680 680 682 683 684 684 685 685 686 686 686
6 688 689 691 691 692 692 692 694 695 695 697 698 698 699 701 702 703 703 705 706 706 707 708 708 710 711 712 713 714 715 716 717 717 719 719 720 720 721 722 722 723 724 725 725 726 727 729 729 729 730 730 731 732 732 735 736 737 738 7
38 745 745 745 746 746 746 747 748 748 748 749 750 751 752 753 754 754 755 755 757 757 757 759 760 761 762 762 763 764 766 767 768 768 772 773 773 774 774 775 776 776 777 778 778 778 781 783
783 784 784 785 788 790 790 791 791 791 792 793 795 795 795 797 797 797 798 798 798 799 800 801 801 802 802 803 803 803 803 804 805 806 806 806 808 808 808 811 812 812 812 813 813 816 816 817 818 818 819 821 822 823 824 825 825 825
826 826 827 827 829 830 830 831 832 832 833 834 836 836 836 839 840 841 842 844 845 845 845 845 845 846 846 847 850 851 854 855 857 857 857 858 858 858 860 861 862 862 866 866 867 867 870 872 872 873 873 873 874 874 874 875 875 877 87
9 880 884 884 885 886 886 886 887 887 887 888 889 889 890 892 892 893 894 895 896 896 897 900 901 902 903 903 904 904 905 905 905 905 906 907 909 910 913 913 915 916 916 918 919 919 920 921 922 922 922 922 924 924 927 929 930 9
975 975 976 977 978 978 979 980 981 982 983 985 987 987 988 988 990 991 992 993 993 993 995 995 996 998 998 998 999 Displayed
Deleted
```

## 第二题代码

```
#include <iostream>
#include <algorithm>
using namespace std;

void Display(int* p,unsigned n) {
    for (int i = 0; i < n; i++) {
        cout << p[i] << " ";
    }
    cout << endl;
}

int digitReverse(int num) {
    int reversedNum = 0;
    int pow10 = 100;
    while (num != 0) {
        int digit = num % 10;
        reversedNum += digit * pow10;
        pow10 /= 10;
        num /= 10;
    }
    return reversedNum;
}

void SortArray(int* p, unsigned n) {
    for (int i = 0; i < n; i++) {
        int num = p[i];
        num = digitReverse(num);
        p[i] = num;
    }
    sort(p, p + n);
}

void DestroyArray(int* arr) {
    delete arr;
}

int main()
{
    int groupNo;
    while (cin >> groupNo) {
        //二维矩阵 result, 行为组数目, 列为数据个数
```

```
int GroupNo = groupNo;
int** result = new int*[GroupNo];
int* DataNo=new int[GroupNo];
while (groupNo--) {
    int dataNo;//当前组数据个数
    cin >> dataNo;
    DataNo[groupNo] = dataNo;
    result[groupNo] = new int[dataNo];
    int* input = new int[dataNo];
    int* sorted = new int[dataNo];//特别排序后的结果
    for (int i = 0; i < dataNo; i++) {
        cin >> input[i];
        sorted[i] = input[i];
    }
    //仅按照定义顺序排列 sorted
    SortArray(sorted, dataNo);
    for (int i = 0; i < dataNo; i++) {
        for (int j = 0; j < dataNo; j++) {
            if (digitReverse(input[i]) == sorted[j]) {
                result[groupNo][j] = input[i];
            }
        }
    }
}
cout << "The result is:" << endl;
for(int i=GroupNo;i>0;i--)
    Display(result[i-1], DataNo[i-1]);
}
```

## 第二题测试

```
3
3 122 131 133
5 4210 2010 3241 1623 232
4 235 14 268 201
The result is:
131 122 133
2010 4210 3241 232 1623
201 14 235 268
2
3 123 124 151
3 6 111 233
The result is:
151 123 124
111 233 6
|
```

### 第三题第一问代码

```
#include <iostream>
using namespace std;

void InitMatrix(int matrix[2][3], int rows, int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            matrix[i][j] = (i+1)*(j+1);
        }
    }
}

void InitMatrix(int matrix[3][4], int rows, int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            matrix[i][j] = i+j+1;
        }
    }
}

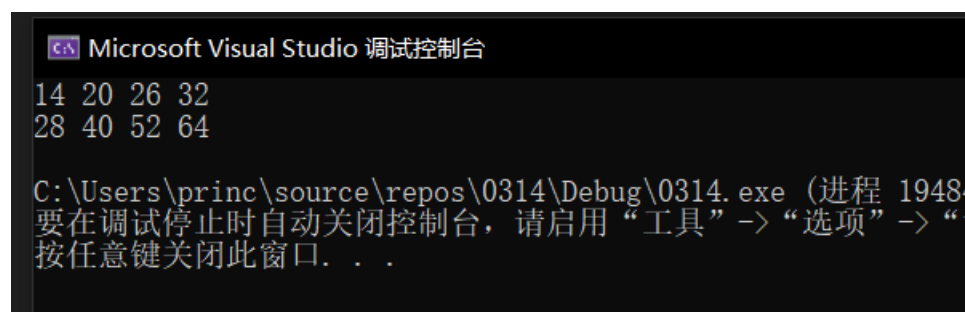
//乘法
int** Multiply(int a[2][3], int b[3][4], int rows_a, int cols_a,
int cols_b) {
    int** c=new int*[rows_a];
    for (int i = 0; i < rows_a; i++) {
        c[i] = new int[cols_b];
    }
    for (int i = 0; i < rows_a; i++) {
        for (int j = 0; j < cols_b; j++) {
            c[i][j] = 0;
            for (int k = 0; k < cols_a; k++) {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }
    return c;
}

// 显示矩阵
void Display(int** matrix, int rows, int cols)
{
    for (int i = 0; i < rows; i++){
```

```
        for (int j = 0; j < cols; j++){
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
}

int main()
{
    const int n = 2;
    const int k = 3;
    const int m = 4;
    int a[n][k], b[k][m];
    int** c;
    InitMatrix(a, n, k); //初始化矩阵 a
    InitMatrix(b, k, m); //初始化矩阵 b
    c = Multiply(a, b, n, k, m); //计算 aXb，实际设计可以省略一个
参数 k
    Display(c, n, m); //输出计算结果，有可能还需要其他参数，根据自己
定义的 c 调整
}
```

### 第三题第一问测试



```
Microsoft Visual Studio 调试控制台

14 20 26 32
28 40 52 64

C:\Users\princ\source\repos\0314\Debug\0314.exe (进程 19484)
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试时自动关闭控制台”。
按任意键关闭此窗口. . .
```

### 第三题第二问代码

```
#include <iostream>
#include <cstdlib>
using namespace std;

// 初始化矩阵
void InitMatrix(int** matrix, int rows, int cols) {
    srand(time(NULL));
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            matrix[i][j] = rand() % 10;
        }
    }
}
```



```
    }  
}  
  
//乘法  
int** Multiply(int** a, int** b, int rows_a, int cols_a, int  
cols_b) {  
    int** c=new int*[rows_a];  
    for (int i = 0; i < rows_a; i++) {  
        c[i] = new int[cols_b];  
    }  
    for (int i = 0; i < rows_a; i++) {  
        for (int j = 0; j < cols_b; j++) {  
            c[i][j] = 0;  
            for (int k = 0; k < cols_a; k++) {  
                c[i][j] += a[i][k] * b[k][j];  
            }  
        }  
    }  
    return c;  
}  
  
// 显示矩阵  
void Display(int** matrix, int rows, int cols)  
{  
    cout << "Displaying" << endl;  
    for (int i = 0; i < rows; i++){  
        for (int j = 0; j < cols; j++){  
            cout << matrix[i][j] << "\t ";  
        }  
        cout << endl;  
    }  
    cout << "Displayed" << endl;  
}  
  
int main()  
{  
    int n, k, m;  
    int i, j;  
    cout << "row of matrix a:" ;  
    cin >> n;  
    cout << "column of matrix a and row of matrix b:";  
    cin >> k;  
    cout << "column of matrix b:";  
    cin >> m;
```

```

int** a = new int*[n];
for (i = 0; i < n; i++) {
    a[i] = new int[k];
}
int** b = new int*[k];
for (i = 0; i < k; i++) {
    b[i] = new int[m];
}
int** c = new int*[n];
for (i = 0; i < n; i++) {
    c[i] = new int[m];
}
InitMatrix(a, n, k); //初始化矩阵 a
Display(a, n, k);
InitMatrix(b, k, m); //初始化矩阵 b
Display(b, k, m);
c = Multiply(a, b, n, k, m); //计算 aXb , 实际设计可以省略一个
参数 k
Display(c, n, m); //输出计算结果, 有可能还需要其他参数, 根据自己
定义的 c 调整
}

```

### 第三题第二问测试

```

row of matrix a:4
column of matrix a and row of matrix b:7
column of matrix b:9
Displaying
3      0      6      7      8      2      2
6      2      2      5      9      7      3
5      3      0      2      9      1      1
7      1      0      0      2      8      2
Displayed
Displaying
3      0      6      7      8      2      2      6      2
2      5      9      7      3      5      3      0      2
9      1      1      7      1      0      0      2      8
2      8      0      6      3      5      0      4      0
2      5      2      2      2      5      7      2      2
1      0      0      3      8      0      1      2      4
0      8      7      8      4      2      4      0      3
Displayed
Displaying
95     118     54     143     91     85     72     78     84
75     121     95     163     157    98     100    92     87
44     84      82     97      85     82     87     58     41
35     31      69     100    135    33     47     62     58
Displayed

```