

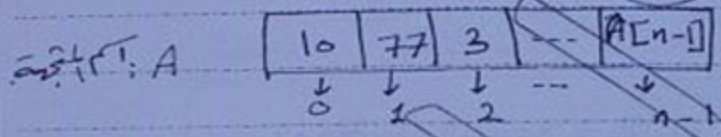
القسم: الرياضيات السنة: الأولى المحاضرة: الأعداد

المادة: جبر خطي. التاريخ: 15/.../... 2014 الدكتور: سمير معطر

المصفوفات والمصفوفات (Array Matrix) :

المصفوفة هي مجموعة من العناصر مرتبة في شكل معين. يمكن اعتبارها مجموعة من العناصر مرتبة في شكل معين.
 نفس كدها ثابت يسمى بعد المصفوفة

« ملاحظة: عندما يأخذ المصفوفات قيم مرتبة في شكل معين في الذاكرة مكانه.
 مقسم كما في الشكل التالي » :



* كيف نحيز بين الخانات :

ان نرقم الخانات ونبدأ بالترقيم من الصفر وحتى $n-1$ مثلا.
 و نضع لثابت الخانة به ليصل المصفوفة

دليل المصفوفة: لتحديد ترتيب عناصر المصفوفة حسب أبعادها.

أي: الخانة 0 قومي المصفوفة الأولى بالمصفوفة.
 " " " " " الثاني " "

أي: $A[0] = 10$ و ذلك:

لا يوجد مصفوفة $A[n]$.
 يتم التعامل مع كل عنصر من ذلك:

أي: اسم المصفوفة - ترتيبها
 (1)

كيف نستطيع تعريف مقبول من نوع متغيرة في C++
في [بعد المتغيرة] لها احدى متغيرات لها نوع عناصر المتغيرة.
عدد عناصرها. متراخ

مثال: تزيد المقدم عن متغيرة ذات بعد 10 اعداد صحيحة

int A[10];

او تزيد المقدم عن متغيرة ذات بعد (50 حرف)

int char C[50];

للمقدم عن متغيرة ذات بعد 5 اعداد حقيقية نكتب:

double d[5];

اما اذا كانت المتغيرة من نوع البوليني فنكتب

bool b[7];

صحة بعد المتغيرة هو 7.

أخيرا وثالثة:

int n = 7;

int x[n];

للمتغير الكتابة بالعدد السابق لانشاء هذه الحالة تكون قد

عرفنا المقبول n وليس متغيرة.

و نكتب كونهما عن ذلك:

const int n = 7;

int x[n];

لقيم المتغيرة [5] اذ فئة اجزاء
A[0]
A[1]
A[2]
A[3]
A[4]


```

#include <iostream.h>
int main() {
char c[100]
int n, i;
do {
cin >> n;
} while (n > 100 || n <= 1);
for (i = 1; i < n; i++)
cin >> c[i];
for (i = 1; i < n; i++)
if (i % 2 != 0)
cout << c[i];
cout << c[1] << c[n-2];
char min = c[0];
int l = 0;
for (i = 1; i < n; i++)
if (c[i] < min) {
min = c[i];
l = i;
}
cout << min << l;
char x;
cin >> x;
bool t = false;

```

```

for (i = 0; i < n; i++)
    if (c[i] == x) t = true;
if (t == true) cout << x << " is found";
else cout << " is not found";
for (i = n - 1; i >= 0; i--)
    cout << c[i];
return 0;
}

```

ملاحظة:

إذا كانت bool غير موجودة في نسخة البرنامج فتكتب بدلاً منها
 enum bool { false = 0; true };

فائدة استخدام النوع البولوي:

لو كان لدينا متغيرة ذات بعد n وجميع قيمها A (محملة)
 $A^* [A, A, A, \dots, A]$

في ترتيب الذاكرة عن A مضافا إذا كانت موجودة A .

طريقة استخدام لإختبار القيم جميعها من قيمة إلى المطلوب.

يمكنها تلخيص true (أي أن القيم المرادة موجودة).

أما لو استخدمنا النوع char فإنه يوضح A مرة A كل

سيفتح كل عنصر ويقوم بطباعة A كل ما وجدها أي

هذا يوضح A مرة A عن شأنته التقنية.

المصفوفات

المصفوفة: هي مجموعة تجميعات مرتبة عام.
ولمعد المصفوفة دائماً يكون (2) وهما للأعمدة والأسطر.
تعريف (c): هي عبارة عن نسبة برمجية مقارنة ثنائي
عدد من الأسطر ط ومعد من الأعمدة m (بعد إجرائها ثابت)
اذن عدد عناصر المصفوفة هو $m \times n$.

تعريف المصفوفة:

[عدد الأعمدة] [عدد الأسطر] اسم المصفوفة نوع عناصر المصفوفة

char c [5][3];

int M [2][3];

دليل الأسطر 0, 1 دائماً تبدأ من الصفر.
الأعمدة 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000.

وبالتالي فإن عدد عناصر المصفوفة:

int M [2][3]

ص: 6 حيث: M [0][0] M [0][1] M [0][2] M [1][0] M [1][1] M [1][2]

M [0][0] M [0][1] M [0][2] M [1][0] M [1][1] M [1][2]

for (i = 0; i < 2; i++)

for (j = 0; j < 3; j++)

cin >> M [i][j];

i = 0

يختبر عندما

التفصيل

j = 0 → cin >> M [0][0]

j = 1 → cin >> M [0][1]

(6)

$i=0, j=2 \rightarrow \text{cin} \gg M[0][2]$
 $i=1, j=0 \rightarrow \text{cin} \gg M[1][0]$
 $\text{'' } j=1 \rightarrow \text{cin} \gg M[1][1]$
 $\text{'' } j=2 \rightarrow \text{cin} \gg M[1][2]$

مثال: أكتب برنامج بلغة ++C يقوم بإدخال مصفوفة
 أبعادها 5x5 من الأعداد الصحيحة ثم يغير التالي:
 1 - حساب مجموع عناصر المصفوفة وطولها
 2 - حساب مربع عناصرها الربيع
 3 - حساب عناصر قطرها الرئيسي والثانوي وطولها
 4 - طباعة المصفوفة على شكل مصفوفة.

```

#include <iostream.h>
int main() {
  int A[5][5];
  int i, j;
  for (i=0; i<5; i++)
  for (j=0; j<5; j++)
    cin >> A[i][j];
  int Sum;
  for (i=0; i<5; i++)
  for (j=0; j<5; j++)
    Sum = Sum + A[i][j];
  cout << Sum;
}

```

عبارة معرفت المصفوفة
 أصبح بالإمكان ادخالها

تكرر هذا البرنامج
 في كل مصفوفة

* تعتبر المصفوفة من المتتالية من الأعداد و ترتيب مجموعها Sum

```
int Sum1 = 0;
for (i = 0; i < 5; i++)
    Sum1 = Sum1 + (A[i][j] * A[i][i]);
cout << Sum1;
int Prod = 1;
for (i = 0; i < 5; i++)
    for (j = 0; j < 5; j++)
        if (i + j == 5 - 1) Prod * = A[i][j];
cout << A[i][j] << " ";
cout << endl; }
return 0; }
```

انتهت الاضافة السابق