

1838 - I2P(I)2019_Yang_CS_hw10

Scoreboard (/contest/scoreboard/1838/)

Time

2019/11/19 21:00:00 6days, 13:32:22 2019/11/26 12:00:00

Clarification

#	Problem	Asker	Description	Reply	Replier	Reply Time	For all team

Overview

Problem ▾

11711 - Dynamic 3D array

Status (/status/?pid=11711) | Limits

Submit (/users/submit/11711)

Description

In this problem, you are asked to design two functions

1.

```
unsigned*** new_3d_array(unsigned n,unsigned m,unsigned k);
```

malloc an $n*m*k$ 3D unsigned array, and then return its address. The main function will check the correctness of your array.

2.

```
void delete_3d_array(unsigned ***arr);
```

Free the memory space of your array that was previously allocated by using malloc. Be careful about the memory uage of your program allocated dynamically so as to avoid MLE.

The two functions have been declared in function.h, and you are asked to complete the function definitions in function.c.

Your program should construct the 3D array by using only three malloc function calls. Notice that malloc is a time-consuming operation.

Note: for OJ submission:

Step 1. Submit only your function.c into the submission block. (Please choose **C** compiler)

Step 2. Check the results and debug your program if necessary.

Input

Please refer to the main function.

The input only has one line, consisting of five positive integers t,n,m,k,r separated by space characters, where t is the number of tests, (n,m,k) represents the array size, and r is a parameter for testing.

Note that $n*m*k \leq 10000000$ and $t*n*m*k \leq 60000000$

Output

In order to test your array's correctness, we will use your array to do some computations and output the results.

Sample Input

Download (data:text/plain;charset=utf-8,60%20100%20100%20100%207122%0A)

60 100 100 100 7122

Sample Output

Download (data:text/plain;charset=utf-8,4266695216%203582066263%202355184006%202590080585%201272726168%0A2727168730%203830671537%203351783560%201832264731%20556185174%2067680956%2022275731%20738670346%20615250193%201198139040)

4266695216 3582066263 2355184006 2590080585 1272726168
2727168730 3830671537 3351783560 1832264731 556185174
2067680956 122275731 738670346 615250193 1198139040

619765246 128111605 1345966600 2044704299 3912466642
3704902008 527967819 823687610 505726373 1820019208
1414007314 1341387693 3919987556 3841906227 3432993290
296141736 3333582015 4197233478 1922513785 2948217552
4139535146 642055013 1824080492 1477484347 4248859214
2970927116 3603540783 1875440426 3308280333 2913600864
1968132446 2179765217 914786072 597221263 1662731338
1636277424 15841719 1330949846 837644417 981255616
1193012966 729149693 571757304 1787542767 4197701398
429582980 2520333543 1135685958 2218180073 1168142812
4188192686 4173466953 159513176 2422641603 1818468330
3295686508 2802786027 2133846194 721639269 2102204016
3236959662 1151454837 635980180 3315448935 506257526
3154988752 9640455 441198786 532734717 1995814748
3492638398 264797657 129656312 2766896975 1725287662
688458144 1812607507 1161089378 3226861161 2136491104
3759358122 2240142577 3992407876 3869112839 1963998598
3975296388 4164577607 1425698626 4199333641 186095964
4178131954 3656316813 2374405808 3481325211 3863154846
2509751596 260622839 1428785526 2346961741 3794079144
3687571878 1190672609 3617851716 1253477591 3582802042
3853947952 2440949935 2508192970 1439958789 3324233824
4026472682 1754495785 1917851700 3948770107 1837477442
1454369868 456501183 2001770910 2716909157 1177367712
3984147818 2827375657 4012301444 883946767 1704627094
3693868272 1118504291 4134521822 3595357421 799916364
1267014114 4047079081 3093581116 2155740955 533781774
2494923540 1008533599 1297618482 3892028681 1796082252
3049616322 399744369 3260173272 2407124415 2052321614
496898980 1819176495 186683190 1422182433 651504640
3522743578 3635544649 1115082308 3374507339 3384023258
3202810096 4199151895 1383578702 2814722461 2281935412
4216588046 2514247893 1330680588 861674295 266681794
3207715016 2740311915 3716101962 3717150977 2013299384
3330607474 3866405057 1726361400 3835150255 680177958
3990480360 1424010835 1184827386 1555131725 491986444
1156613818 3977883285 7252408 1213602275 1432261966
2516106864 3627655807 1252820050 883934405 824682136
3101707326 385021057 408541328 3843575203 3647132006
3625813912 2554462847 1444387414 830805189 1805649028
3964423418 1589462125 3109835076 3701364159 3100773150
1365607712 2117708019 4176481354 4084617105 2993444176
1357547286 4028318785 3798981048 1162761799 2804275554
2880720072 2807261375 1758237490 1920688757 2767392168
3929112862 1296996957 3544643468 1962896903 3842308434
460442228 1015703579 3513222474 3251724045 2075938220
432549826 2512435253 3945705768 3758359939 186421714
351451712 2095775411 1239668302 2438943373 226142856
3718512922 1203918325 1108566820 470900227 782119730
1974734136 2010178143 3265294618 2928802665 1204805160
1039155002 2791685317 3135552864 3795827271 4289549550
260077332 626548219 161978526 3731300061 1354444636
3635831502 945521505 815928896 4211420971 2134652190
4157251912 557837859 97418294 1391060261 2600145436
175562210 2867240613 175515148 2589777511 1190753394
784144696 3605144147 736706026 3112849117 246040004
1678818830 2195495393 4284430840 2808878699 883730178

Partial Judge Code

11711.c (/problem/partial/11711.c/)

Partial Judge Header

11711.h (/problem/partial/11711.h/)

Discuss