**Notatka z 11.01.2017r**

{

char a 1

int b 4

char c 1

} razem 12 bajtów

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | x | x | x | b | b | b | b | c | x | x | x |

x dopełnia puste miejsca

{

int a;

double b;

} QQ; razem 16 bajtów

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | a | a | a | x | x | x | x | b | b | b | b | b | b | b | b |

QQ tab[20][30]

5000 początek

tab[0][0] = 5000 – 5015

tab[0][0].a = 5000 – 5007

tab[0][0].b = 5008 – 5015

tab[5][10] = 5000 + (5\*30+10)\*16 -> 7560 – 7575

element tab[i][j]

ogólnie tab[m][n]

adres + (i \* n + j) \* rozmiar struktury -> do końcowego trzeba dodać rozmiar struktury

Wzór na trójwymiarową

Info odnośnie kartkówki (60min)

30 minut liczenie adresów na kartkach itp. (8pkt)

30minut programowanie na komputerach (~7pkt) funkcja albo mpi