Memory Map of 1D array of pointers to strings

char \*names[5] = {"manas","om","nihar","abhiram","ajay"};

Here 1st the compiler allocated continuous memory locations for 5 names & then returned their addresses to pointer array at the left.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| manas | om | nihar | abhiram | ajay |

600 607 610 617 625

Memory Adr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 600 | 607 | 610 | 617 | 625 |

names[5] 🡪

However this concept can be used with self initialised array of strings only. It should not be used wherein we are going to accept the strings from user at runtime. This is bcoz, if we only make the following declaration

char \*names[5];

Then this will create an array of 5 pointers to strings with some values in array which are called as “garbage values”. If these garbage values are used to point to strings accepted from user, we may get incorrect results.

om(820) nihar (901) abhi~~ram~~(1024)

ajay(1028)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 600 | 820 | 0 | 1024 | 1028 |

manas(600)

names[5] 🡪

Result 🡪 manas om null abhiajay ajay (Corrupted data)