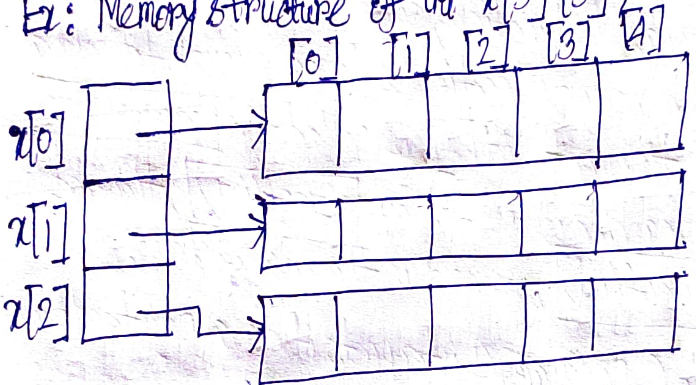
• MALLOC(list, n*sizeof(int))

2.2.2 Two-Dimensional Array

• A two-dimensional array is represented as a one-dimensional in which each element is, itself, a one-dimensional array.

Ex: Memory structure of int x[3][5];



• C finds the element x[i][j] by first accessing the pointer in x[i]. This pointer gives us the address, in memory, of the zeroth element of row i of the array. Then by adding j*sizeof(int) to this pointer, the address of the jth element of row i (i.e. element x[i][j]) is determined.

```
int** make2dArray(int rows, int cols)
{ /* create a two-dimensional rows x cols array */
    int **x, i;
   /*get memory for row pointers */
   MALLOC(x, rows * sizeof(*x));
  /* get memory for each row */
  for(i=0; i<rows; i++)
      MALLOC(x[i], cols*sizeof(**x));
  return x;
}
```

```
#define MALLOC(P, S) \
if(!((P)=malloc(S))){ \
fprintf(stderr, "Insufficient
          memory"); \
exit(EXIT_FAILURE); \
}
```

• The function $\boxed{\text{calloc}}$ allocates a user-specified amount of memory and initializes the allocated memory to 0 (i.e. all allocated bits are set to 0); a pointer to the start of the allocated memory is returned. In case there is insufficient memory for allocation, NULL is returned.