|  |  |
| --- | --- |
| 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 | #include <stdio.h>  #include <stdlib.h>  #include <math.h>    int main( int argc, char\* argv[] )  {        // Size of vectors      int n = 10000;        // Input vectors      double \*restrict a;      double \*restrict b;      // Output vector      double \*restrict c;        // Size, in bytes, of each vector      size\_t bytes = n\*sizeof(double);        // Allocate memory for each vector      a = (double\*)malloc(bytes);      b = (double\*)malloc(bytes);      c = (double\*)malloc(bytes);        // Initialize content of input vectors, vector a[i] = sin(i)^2 vector b[i] = cos(i)^2      int i;      for(i=0; i<n; i++) {          a[i] = sin(i)\*sin(i);          b[i] = cos(i)\*cos(i);      }        // sum component wise and save result into vector c      #pragma acc kernels copyin(a[0:n],b[0:n]), copyout(c[0:n])      for(i=0; i<n; i++) {          c[i] = a[i] + b[i];      }        // Sum up vector c and print result divided by n, this should equal 1 within error      double sum = 0.0;      for(i=0; i<n; i++) {          sum += c[i];      }      sum = sum/n;      printf("final result: %f\n", sum);        // Release memory      free(a);      free(b);      free(c);        return 0;  } |
|  |  |

Vector addition