```
// size specified by user
/* Host Memory Initialization */
h input1=(float*)malloc(sizeof(float)*size);
h input2=(float*)malloc(sizeof(float)*size);
h output=(float*)malloc(sizeof(float)*size);
init data(h input1,size);
init data(h input2,size);
/*Device Buffer Creation */
buf input1 = clCreateBuffer(clContext,CL MEM READ WRITE,size*sizeof(float),NULL,&clStatus);
buf input2 = clCreateBuffer(clContext,CL MEM READ WRITE,size*sizeof(float),NULL,&clStatus);
buf_ouput =clCreateBuffer(clContext,CL_MEM_READ_WRITE,size*sizeof(float),NULL,&clStatus);
/*H2D Copy */
clEnqueueWriteBuffer(clCommandQueue,d_A0,CL_FALSE,0,size*sizeof(float),buf_input1,0,NULL,NULL);
clEnqueueWriteBuffer(clCommandQueue,d_Anext,CL_TRUE,0,size*sizeof(float),buf_input2,0,NULL,NULL);
/* Setting Kernel Parameters */
clSetKernelArg(clKernel,0,sizeof(float),(void*)&buf input1);
clSetKernelArg(clKernel,1,sizeof(float),(void*)&buf input2);
clSetKernelArg(clKernel,2,sizeof(cl mem),(void*)&buf output);
 /*Work-item Processing*/
 grid[3]=\{size,0,0\};
 clEnqueueNDRangeKernel(clCommandQueue,clKernel,2,NULL,grid,block,0,NULL,NULL);
 /*D2H Copy*/
clEnqueueReadBuffer(clCommandQueue,buf_output,CL_TRUE,0,size*sizeof(float),h_output,0,NULL,NULL);
```