```
-- 96bal _ void find_roots_kernel (float *A, float *B, float *C, float *XO,
                                           float * x1, float *fx0, float *fx1, int n) {
     int row = (blockDim. J)* (blockIdx &) + throadIdx. &i
      int col = (blockbirm.x)*(block idx x)+ thread idx. x;
      int id = (8 FidDim X) * ( block Dim X) * row + col)
      fload di
       d = sart (Brid] + Brid] - 4+ Arid] * (Cid]) 1// Jb= 4ac = 2101.
      XO[id] = (-BCid]@d) / [2 * A[id]) [ 기항상 서로 다는 두 실근은 가진다는 전제가 있으므로 d>oolch.
      X1 [id] = (-B(id)(d))/(2*A[id]); 11 -6-18-90 < -6+16-90 0103 1-13 (25) Ext.
      FXO[id] = (A[id] * XO[id] + B[id]) * XO[id] + C[id]; " ax + bx + (=0 =1 4 and xoll plant + it
      FX[[[d] = (A[[d]*XI[[d]+B[[d])*XI[[d]+C[[d]]1/22 CHSter its con user iter.
3
void find-roots_GPU (float *A, float *B, float *E, float *XO, float *XI, float *FXD, float *FXI
     floot * d-A, *d-B, *d-C, *d-XO, *d-XI, *d-FXO, *d-FXI)
     size + sizei
      Size = N* size of (floot) ; "derice Memory = 312 x12
      CUDA_CALL ( Cuda Malloc (&d_A, size)), 11 device Memory its
      CUDA_CALL (cuda Memopy (d_A, A, size, cuda Memopy Host To Device ), 11 host Memory original at
                                                                       device Memory 2 4xt.
      CUDA_CALL (cuda Mallocc&d_B, STZe)); // device Memory 25
      CUBA_CALL (cuda Memapy (d-B, B, Size, and a Memopy Host To Device)) Il host memory HIPE its
      CUDA-CALL ( Cuda Malloc ( &d-C, Size)): //device Memory ofty
                                                                         device Memory 3 4xt.
      CVDA_CALL (cuda Mencpy (d.c, c, size, cuda mem cpy Hostro Device)), 1 host memory unger its
                                                                        device memory 3 gxt.
      CUDA-CALL ( cuda Malloc (&d_XO, 5128)) 1/1/1/1/1/1/1/
      CUDA_CALL (cuda Malloc(ad_XI, STZE)) i // device Memory //
      CUDA_CALL(CUMaMalloc(&d_FXO, 5720)); //
      int n=Block_size; int nz=n/Block_size;
       dim3 dimBlock (Block-SIZE, BLOCK-SIZE),
       dim3 dimerid ( n1/dimBlock, x, n2/dimBlock y);
       CHECK_TIME_INIT_GPU() CHECK_TIME_START_GPUX) // GAM 1/25/28
       Find_roots_ Kernel << dimGtid, dimBlock >>> (d-A,d-B,d-C,d_X0,d_X1,d_FX0,d_EX1); // AZ FOO
        CHECK_TIME_END_GPU(device_time) CHECK_TIME_DEST_GPU() // GPU KILLY
        CUDA-CALL ( CUDA MEMCPY ( XD, d-XD, STZE, CUDAMEMCPY DEVILE TO HOST) 11 11 11 11 11 11 11 11
        CUDA_CALL(Condamoncpy(X), d-X1. Size, and Momory Device To Host))) / Zertis Host
        CUBA-CALL(Cuda Memcpy (FXD, d-FXD, 5728, Cuda Memcpy Device To HOCT)) 1/1 mamory 2 & Atot.
         CUDA_CALL ( and a Memopy C FXI, d - FXI, STZE, cuda Memory Device To Ho H ) II 11 11 11 11 11 11 11 11 11 11
```

```
-- dlobal _ void Fibonacci _ kernel(int *x, int *8) {
    int row = (block Dim. 4) * ( block Idx. 4) + thread id & . 4)
    int col= (blockDim x1) * (blockIdx.x1)+ thread Ida, x;
    float sqrt-5 = sqrt(5,0); 1155
    floor topo=1:0, top1=1.0; foot X0, K-1;
     X0=(1+59++5)/2.0] X-1=(1-59++-5)/2.0]
     for( int == 0; 7 < x (id); i++) }
        tmpo *= x-0; tmp1 *= x-1; 11 tmpo=(1+15)xid, tmp1=(1-15)xid
     3[id] = (int)((+mpo-+mp())sqrt-5+0.5); // INLANT Age id HOW of
3
 Cuda Error - + Fibonacci - GPU (int xx, int x 4) {
       cuda error_t cuda status = cu da set Device co);
       int *dx, *d-4;
       Size -t size = N+size of CTINE) TI device Hemory 却是 지정.
       THAT N = BLOCK_SIZE, NZ = n/BLOCK_SIZE;
        CUDA_CALLCCUDA Malloc (&d_x, Size)); Idevice Memory 25
        CUDA_CALL ( CUDA Mem CPY (d. x, size, Cuda Mem cpy Host To Device)); // host Memory 21 215
        CUPA_CALL (cuda Malloc(&d_y, size)); // device Memory sty.
                                                                       device Hemory z tht.
        dim3 dimBlock(BLOCK_SIZE, BLOCK_SIZE);
         dim3 dimGHd(n1/dimBlock.x. n2/dimBlock.y);
        CHECK_TIME_INIT_GPUCTI // GPU NOTING
        CHECK_TIME_START_GADULII/ GAU 시간측정
        Fibonacci - Kernel ( dim Grid, dim Block >>> (d-X,d-y) i // +12 +34
         CHECK-TIME_ END_GADUCCELICE_ Fine) i 1 GADU AIXENT
        CHECK_TIME_DEST_GPUC ) TIGPUNTEN
         CUBA-CALL(Cudaberice Synchronize(1) 1 / 커널이 끝박때까지 기다짐, 동기화를 위한 것.
         CUDA_CALL (cuda Memcov (Y, d_Y, size, cuda Memcovo evice To Host ) 1/ MILE ELEVATI (===
         neturn cudastatusi
                                                                         Host Memoryz &ctit
 3
```

```
- 410bal _ void Sum_n elements_kernel (int *d_Array, int *d_sum, int N, int Nf) {
      int row = (block Dim. y) *( block Idx. y) + thread Idx y;
      int col = (block Dim.x) * (block Idx.x) + thread. Idx. x;
      intid = cgridDim. xxx (blockDim x) x row + coli
       for(int i=0; i<Nf; i+1){
                                                      / Tol 비재 원스를 기준으로 왼쪽 원소 NF개를 더한다.
          「「はってンニロ」 d_SUM CTO」ナー d_Array ETO-17711 を致 のはひ チをはり 幸也
     · for (int i=1 = T < NF = T+1) {
                                        11 प्रमाम सिर्ट 11803 व्हेंसिर (NF-1) यह अरूप
          「f(id+i< N) d-Sum Eid] t= d-Array[idti] 1 作動 MUZI チンのに 新り
        11 主要なる d-Sum cid) = ごtht d-Array[k] e1 またき さたてた
 3
 CUDATION + SIM_N_ Elements_GPUCTH *P_Array. Int *P-sum, Int NF) &
    Cuda Error_t cudastatus = cudas et Device(0);
     int *d_Array, *d_Sym;
    Size + Size = N * Size of (int) i / device Memory = 1 77
     int n = BLOCK-GIZE, nz= N/BLOCK-SIZE;
     CUDA_CALL CCUDANAMORCES d. Array, Size)) // device Nemony its
      CUDA_CALL (CUdaMemcay(d_Array, >-Array, size, cutaMemcpyHostio De vice)); // host Memory=1 25
      CUDA_CALL (cuda Malloc ( &d_Sum, size ) i // device Memory of
                                                                           device Hemory 3 44
      dim3 dimBlock CBLOCK_SIZE, BLOCK_SIZE)]
      dim3 dimarid (n1/dimblock.x, n2/dimblock.4);
       CHECK_TIME_INIT_GOULDS 11 11 11 11 11
       CHECK_TIME_START_GRAVEST 11 GAN NITED 11
       sum-n-clements-kernel «cdimerid, dimblock »> (d-Arry, d-sum, N, Nf); 1/4/2 43/4
       CHECK_TIME_END_GPU(device_time) I GPU (124)
       CHECK_TIME_BECT_GPUC);
                                          11 GPU NIT = TI
       CUDA_CALL (CURDEVICESYNCHANIZECT) 1 HEOL BYTHINH NOTE
       CUDA_CALL Couda Mamcay (P-sym, d-sym, site, cuda Memopy Device To HOST)) 1/1 对性 块哲明 行門 항은
       return cudactatus;
                                                                            HOST MENDON'S RYTEL
```