

PPL WEEK6 SUBMISSION

Name: Davasam Karthikeya Section: AIMLB Reg No: 230962326

1. Write a program in CUDA which performs convolution operation on one-dimensional input array N of size $width$ using a mask array M of size $mask_width$ to produce the resultant one dimensional array P of size $width$

Code:-

```
#include <cuda.h>
#include <stdio.h>

__global__ void convolution1D(int *N, int *M, int *P, int width, int mask_width)
{
    int i = blockIdx.x*blockDim.x + threadIdx.x;
    if(i > width)return;

    float pval = 0;
    int start = i - (mask_width)/2;
    for(int j = 0; j < mask_width; j++){
        if(start + j >= 0 && start + j < width){
            pval += N[start + j]*M[j];
        }
    }
    P[i] = pval;
}

int main(){
    printf("-----Davasam Karthikeya, 230962326-----\n");

    int width, mask_width;
    printf("Enter width of Input arr: ");scanf("%d", &width);
    printf("Enter width of Mask arr: ");scanf("%d", &mask_width);

    int N[width], M[mask_width];
    for(int i = 0; i < width; i++) N[i] = 2 * i + 1;
    for(int i = 0; i < width; i++) M[i] = i;

    int *dM, *dN, *dP;
    cudaMalloc((void **)&dM, sizeof(int) * mask_width);
    cudaMalloc((void **)&dN, sizeof(int) * width);
    cudaMalloc((void **)&dP, sizeof(int) * width);

    cudaMemcpy(dM, &M[0], sizeof(int) * mask_width, cudaMemcpyHostToDevice);
    cudaMemcpy(dN, &N[0], sizeof(int) * width, cudaMemcpyHostToDevice);

    convolution1D<<<(width-1)/256 + 1, 256>>>(dN, dM, dP, width, mask_width);
    int P[width];
    cudaMemcpy(P, dP, sizeof(int) * width, cudaMemcpyDeviceToHost);

    for(int i = 0; i < width; i++)printf("%d ", P[i]);
    printf("\n");

    cudaFree(dM);
    cudaFree(dN);
```

```

    cudaFree(dP);

    return 0;
}

```

Output:-

```

mca@computinglab26-27: ~/230962326/PPL/Week6
(dse) mca@computinglab26-27:~/230962326/PPL/Week6$ nvcc q1.cu -o q1
(dse) mca@computinglab26-27:~/230962326/PPL/Week6$ ./q1
-----Davasam Karthikeya, 230962326-----
Enter width of Input arr: 200
Enter width of Mask arr: 10
500 583 658 724 780 825 870 915 960 1005 1050 1095 1140 1185 1230 1275 1320 1365
1410 1455 1500 1545 1590 1635 1680 1725 1770 1815 1860 1905 1950 1995 2040 2085
2130 2175 2220 2265 2310 2355 2400 2445 2490 2535 2580 2625 2670 2715 2760 2805
2850 2895 2940 2985 3030 3075 3120 3165 3210 3255 3300 3345 3390 3435 3480 3525
3570 3615 3660 3705 3750 3795 3840 3885 3930 3975 4020 4065 4110 4155 4200 4245
4290 4335 4380 4425 4470 4515 4560 4605 4650 4695 4740 4785 4830 4875 4920 4965
5010 5055 5100 5145 5190 5235 5280 5325 5370 5415 5460 5505 5550 5595 5640 5685
5730 5775 5820 5865 5910 5955 6000 6045 6090 6135 6180 6225 6270 6315 6360 6405
6450 6495 6540 6585 6630 6675 6720 6765 6810 6855 6900 6945 6990 7035 7080 7125
7170 7215 7260 7305 7350 7395 7440 7485 7530 7575 7620 7665 7710 7755 7800 7845
7890 7935 7980 8025 8070 8115 8160 8205 8250 8295 8340 8385 8430 8475 8520 8565
8610 8655 8700 8745 8790 8835 10473 11943 13244 14375 15335 16123 16738 17179 1
7445 17535 17625 17715 14196 11060 8309 5945
(dse) mca@computinglab26-27:~/230962326/PPL/Week6$ 

```

2. Write a program in CUDA to perform selection sort in parallel.

Code:-

```

#include <cuda.h>
#include <stdio.h>

__global__ void selectionSort(int *Arr, int *Out, int size){
    int gtid = blockIdx.x * blockDim.x + threadIdx.x;
    if (gtid >= size) return;

    int pos = 0;
    for (int i = 0; i < size; i++){
        if (Arr[i] < Arr[gtid] || (Arr[i] == Arr[gtid] && i < gtid)){
            pos++;
        }
    }
    Out[pos] = Arr[gtid];
}

```

```

int main(){
    printf("-----Davasam Karthikeya, 230962326-----\n");

    int size;
    printf("Enter size of Arr: ");scanf("%d", &size);

    int Arr[size], Out[size];
    for(int i = 0; i < size; i++)Arr[i] = rand() % 1000;

    int *dArr, *dOut;

    cudaMalloc((void **)&dArr, sizeof(int) * size);
    cudaMalloc((void **)&dOut, sizeof(int) * size);

    cudaMemcpy(dArr, &Arr[0], sizeof(int) * size, cudaMemcpyHostToDevice);

    selectionSort<<<(size-1)/256 + 1, 256>>>(dArr, dOut, size);

    cudaMemcpy(Out, dOut, sizeof(int) * size, cudaMemcpyDeviceToHost);

    for(int i =0; i < size; i++)printf("%d ", Out[i]);
    printf("\n");

    cudaFree(dArr);
    cudaFree(dOut);

    return 0;
}

```

Output:-

```

mca@computinglab26-27: ~/230962326/PPL/Week6
(dse) mca@computinglab26-27:~/230962326/PPL/Week6$ nvcc q2.cu -o q2
(dse) mca@computinglab26-27:~/230962326/PPL/Week6$ ./q2
-----Davasam Karthikeya, 230962326-----
Enter size of Arr: 300
11 11 12 19 22 27 29 31 34 36 42 43 58 59 60 62 67 69 71 82 84 84 87 91 94 95 97
97 117 121 123 124 124 128 131 135 139 143 149 163 167 170 172 178 183 190 193
197 198 199 211 219 226 227 228 228 229 235 237 245 258 270 275 276 280 281 282
286 299 301 305 305 306 311 313 315 317 320 324 324 327 335 336 340 340 340 343
348 350 353 355 362 364 365 367 368 368 368 369 370 373 378 379 379 383 386 393
395 399 403 404 407 413 416 421 421 422 426 428 429 432 434 434 437 440 441 443
444 444 451 452 456 464 465 466 467 468 470 474 481 488 490 491 492 492 497 499
500 503 505 505 507 522 526 528 529 530 537 538 539 540 542 545 550 551 555 567
567 568 570 582 584 586 586 590 600 601 605 606 611 613 618 619 622 624 624 626
637 644 649 651 652 660 661 668 669 675 676 683 688 689 690 708 708 709 715 723
725 729 729 730 732 736 736 739 743 743 746 750 754 756 763 763 764 764 771 772
773 776 777 782 784 786 788 793 793 795 796 801 802 804 808 810 813 814 818 818
828 829 840 841 846 846 856 856 857 858 859 862 862 865 871 873 878 881 886 894
895 899 899 902 904 914 915 917 917 919 921 924 925 926 926 927 928 929 930 932
933 936 954 956 959 965 973 980 981 987 996 996 996
(dse) mca@computinglab26-27:~/230962326/PPL/Week6$ 

```

3. Write a program in CUDA to perform odd even transposition sort in parallel.

Code:-

```
#include <cuda.h>
#include <stdio.h>

__global__ void oddEvenSort(int *Arr, int size, int label){
    int gtid = blockIdx.x * blockDim.x + threadIdx.x;
    if (gtid >= size - 1) return;

    if(gtid < size && (gtid % 2) != label)return;

    if(Arr[gtid] > Arr[gtid + 1]){
        int temp = Arr[gtid];
        Arr[gtid] = Arr[gtid + 1];
        Arr[gtid + 1] = temp;
    }
}

int main(){
    printf("-----Davasam Karthikeya, 230962326-----\n");

    int size;
    printf("Enter size of Arr: ");scanf("%d", &size);

    int Arr[size];
    for(int i = 0; i < size; i++)Arr[i] = rand() % 1000;

    int *dArr;
    cudaMalloc((void **)&dArr, sizeof(int) * size);

    cudaMemcpy(dArr, &Arr[0], sizeof(int) * size, cudaMemcpyHostToDevice);
    for(int i =0; i < size; i++){
        oddEvenSort<<<(size-1)/256 + 1, 256>>>(dArr, size, i % 2);
        cudaDeviceSynchronize();
    }

    cudaMemcpy(Arr, dArr, sizeof(int) * size, cudaMemcpyDeviceToHost);

    for(int i =0; i < size; i++)printf("%d ", Arr[i]);
    printf("\n");

    cudaFree(dArr);

    return 0;
}
```

Output:-

```
mca@computinglab26-27: ~/230962326/PPL/Week6
(dse) mca@computinglab26-27:~/230962326/PPL/Week6$ nvcc q3.cu -o q3
(dse) mca@computinglab26-27:~/230962326/PPL/Week6$ ./q3
-----Davasam Karthikeya, 230962326-----
Enter size of Arr: 300
11 11 12 19 22 27 29 31 34 36 42 43 58 59 60 62 67 69 71 82 84 84 87 91 94 95 97
97 117 121 123 124 124 128 131 135 139 143 149 163 167 170 172 178 183 190 193
197 198 199 211 219 226 227 228 229 235 237 245 258 270 275 276 280 281 282
286 299 301 305 305 306 311 313 315 317 320 324 324 327 335 336 340 340 340 343
348 350 353 355 362 364 365 367 368 368 369 370 373 378 379 379 383 386 393
395 399 403 404 407 413 416 421 421 422 426 428 429 432 434 434 437 440 441 443
444 444 451 452 456 464 465 466 467 468 470 474 481 488 490 491 492 492 497 499
500 503 505 505 507 522 526 528 529 530 537 538 539 540 542 545 550 551 555 567
567 568 570 582 584 586 586 590 600 601 605 606 611 613 618 619 622 624 624 626
637 644 649 651 652 660 661 668 669 675 676 683 688 689 690 708 708 709 715 723
725 729 729 730 732 736 736 739 743 743 746 750 754 756 763 763 764 764 771 772
773 776 777 782 784 786 788 793 793 795 796 801 802 804 808 810 813 814 818 818
828 829 840 841 846 846 856 856 857 858 859 862 862 865 871 873 878 881 886 894
895 899 899 902 904 914 915 917 917 919 921 924 925 926 926 927 928 929 930 932
933 936 954 956 959 965 973 980 981 987 996 996 996
(dse) mca@computinglab26-27:~/230962326/PPL/Week6$
```