Date : 19/02/2026

PPL WEEK6 SUBMISSION

Name: Davasam Karthikeya Section: AIMLB Reg No: 230962326

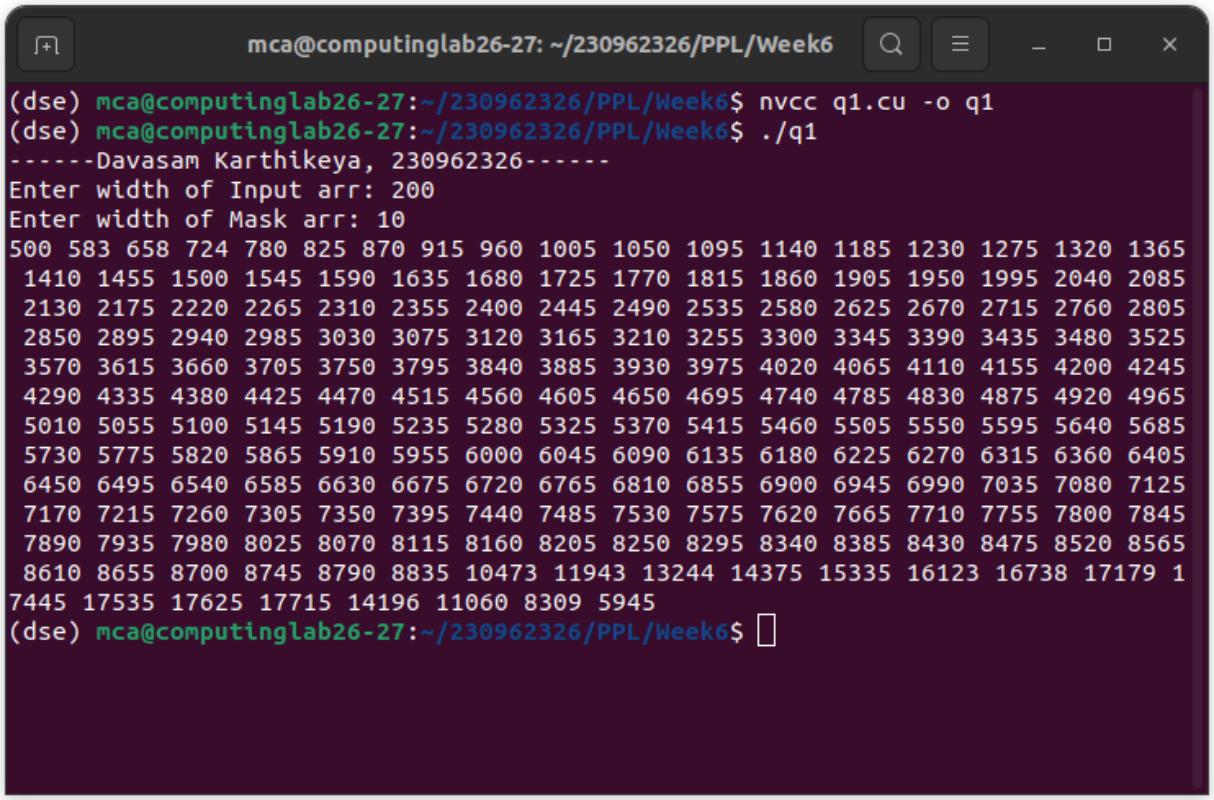
1. Write a program in CUDA which perfonns convolution operation on one-dimensional input arrayN of size width using a mask array M of size mask_width to produce the resultant one dimensional array P of size width

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(width) * width, cudaMemcpyDeviceToHost);

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

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

```
    cudaFree(dP);

    return 0;
}
```

Output:-



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

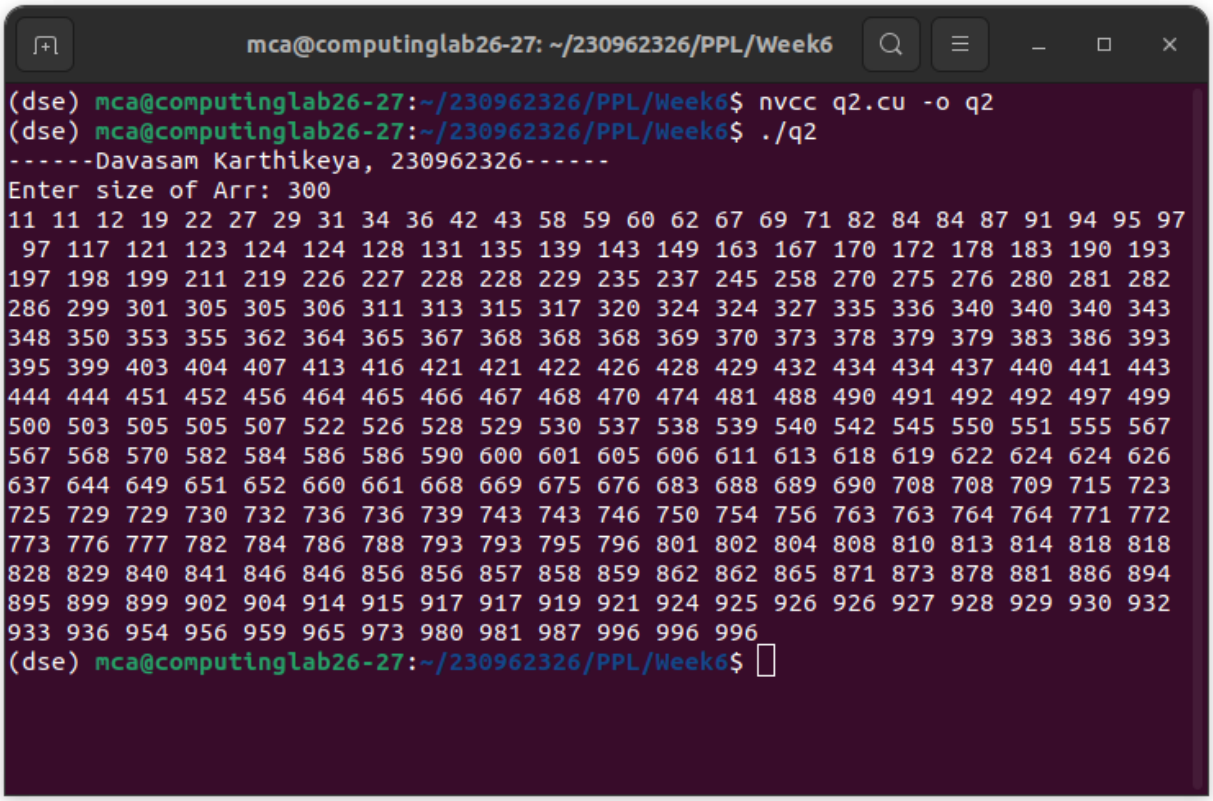
Code:-

```
#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];
}
```

Date : 19/02/2026

```
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:-



Date : 19/02/2026

3. Write a program in CUDA to perforn odd even transposition soryt 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;
}
```

Date : 19/02/2026

Output:-