Computer Science & Engineering Department
I. I. T. Kharagpur

Compilers Laboratory: CS39003

3rd Year CSE: 5th Semester

Assignment - 1 *Marks: 10*

Assignment Out: *22nd July, 2014* Report on or before: *28th July, 2014*

1. Translate the following C program using GCC/Linux to the assembly language program of x86-32 (Intel 32-bit processor).
   `cc -Wall -S` <*file name*>`.c`
   Do not give any optimization option. The *file name* should be `ass1_roll.c` where `roll` is your roll number.

   Write comments in the assembly language code corresponding to the program `<file name>.s`. Comments should explain the corresponding assembly language instructions and also should clearly show the connection between the C program and the assembly language program.

```
/*
 * ass1.c Generate assembly code of x86-32 and comment
 */
#include <stdio.h>
#define ORD 20
void cs(int n, int data[][ORD]);
void po(int n, int data[][ORD], int type, int ind);
int main()
{
    int n, i, j ;
    int data[ORD][ORD];

    printf("Enter the order of the square matrix: ");
    scanf("%d", &n);
    printf("Enter the matrix in row-major order:\n");
    for(i=0; i<n; ++i)
        for(j=0; j<n; ++j) scanf("%d", &data[i][j]);
    printf("The input matrix is:\n");
    for(i=0; i<n; ++i){
        for(j=0; j<n; ++j) printf("%d ", data[i][j]);
        putchar('\n');
    }
    printf("In cs order:\n");
    cs(n,data);
    return 0;
}

void cs(int n, int data[][ORD]){
     if(n == 0) {
          putchar('\n');
          return;
     }
     if(n == 1) {
        printf("%d\n", data[0][0]);
        return ;
     }
     po(n, data, 1, 0);
     cs(n-2, (int (*)[ORD])(&data[1][1]));
}

void po(int n, int data[][ORD], int type, int ind){
     switch(type){
        case 1:
```

```
                if(ind == n-1) po(n, data, 2, 0);
                else {
                        printf("%d ", data[0][ind]);
                        po(n, data, 1, ind+1);
                }
                return ;
        case 2:
                if(ind == n-1) po(n, data, 3, n-1);
                else {
                        printf("%d ", data[ind][n-1]);
                        po(n, data, 2, ind+1);
                }
                return ;
        case 3:
                if(ind == 0) po(n, data, 4, n-1);
                else {
                        printf("%d ", data[n-1][ind]);
                        po(n, data, 3, ind-1);
                }
                return ;
        case 4:
                if(ind == 0) return ;
                else {
                        printf("%d ", data[ind][0]);
                        po(n, data, 4, ind-1);
                }
                return ;
    }
}
```

2. The commented assembly language program should remain syntactically correct.

3. Intel assembly language manual and other reading materials are available at
`http://cse.iitkgp.ac.in/~goutam/`