**מחשוב מקבילי ומבוזר** – (קורס מס' 10324)

***מותר שימוש בכל חומר אין לכתוב בעפרון.***

***אין להשתמש במחשב אישי ובטלפון סלולרי. משך הבוחן: 90 דקות***

***מרצה: ד''ר בוריס מורוז***

***שאלה מס' :1*  (בס''כ 30 נקודות)**

לפניך שלוש תוכנות המבוססות על שימוש ב-MPI עם בעיות מימוש – dead lock, אי התאמה בין תהליכים ואחרות. יש לציין את הבעיה) אין טעיות קומפילציה בקוד(.

יש להניח שבכל הריצות מספר המעבדים הוא **שלוש**.

***שאלה מס' 1.1***

**#include <mpi.h>**

**#define MASTER 2**

**#define SLAVE 1**

**#define SIZE 10**

**int main(int argc,char \*argv[])**

**{**

**int myid;**

**int data[SIZE] = {5, 6, 7};**

**MPI\_Status \*status;**

**if (myid == MASTER)**

**MPI\_Recv(data, 2, MPI\_INT, SLAVE, 0, MPI\_COMM\_WORLD, status);**

**else**

**MPI\_Send(&data[2], 1, MPI\_INT, SLAVE, 0, MPI\_COMM\_WORLD);**

**return 0;**

**}**

***שאלה מס' 1.2***

**#define MASTER 0**

**#define SLAVE 1**

**int main(int argc,char \*argv[]) {**

**int myid, result;**

**MPI\_Status \*status;**

**MPI\_Comm\_rank(MPI\_COMM\_WORLD,&myid);**

**MPI\_Init(&argc,&argv);**

**MPI\_Scatter(&result, 2, MPI\_INT, &result, 2, MPI\_INT, 0, MPI\_COMM\_WORLD);**

**if (myid == MASTER)**

**MPI\_Finalize();**

**return 0;**

**}**

***שאלה מס' 1.3***

**#define MASTER 0**

**#define SLAVE 1**

**int main(int argc,char \*argv[]) {**

**int myid;**

**MPI\_Status status;**

**int numbers[3] = {40, 50, 60};**

**char \*msg = "Programming";**

**MPI\_Init(&argc,&argv);**

**MPI\_Comm\_rank(MPI\_COMM\_WORLD, &myid);**

**if (myid == MASTER)**

**MPI\_Bcast(msg, 3, MPI\_CHAR, 0, MPI\_COMM\_WORLD);**

**else**

**MPI\_Recv(numbers, 3, MPI\_INT, myid - 1, 0, MPI\_COMM\_WORLD, &status);**

**MPI\_Send(numbers, 3, MPI\_INT, myid + 1, 0, MPI\_COMM\_WORLD);**

**MPI\_Finalize();**

**}**

***שאלה מס' :2*  (בס''כ 50 נקודות)**

Four very large arrays of integers A, B, C, D of same size N are initially placed each in the memory of one of four processes: array A on p0, array B on p1, array C on p2 and array D on p3.

Write parallel program that performs following:

Calculates and prints the number of members in these arrays for which

A[i] > B[N - i - 1]

C[i + 1] == D[i -1] ( i is not equal 0 or N-1)

* (15 נקודות) Write a pseudo-code of your solution
* (35 נקודות) Implement your solution with MPI

Notes:

* assume that N is even;
* and you don't need to initialize the arrays A, B, C, D;

***שאלה מס' :3*  (בס''כ 20 נקודות)**

What is an output of the following program:

**#include <omp.h>**

**#include <stdio.h>**

**int main(int argc,char \*argv[]) {**

**omp\_set\_num\_threads(3);**

**#pragma omp parallel for**

**for (int i = 0; i < 4; i++)**

**printf("First i = %d\n", i);**

**printf("--- 1 ---\n");**

**#pragma omp parallel**

**printf("--- 2 ---\n");**

**printf("--- 3 ---\n");**

**#pragma omp parallel**

**{**

**printf("--- 4 ---\n");**

**#pragma omp sections**

**{**

**#pragma omp section**

**printf("--- 5 ---\n");**

**#pragma omp section**

**printf("--- 6 ---\n");**

**#pragma omp section**

**printf("--- 7 ---\n");**

**}**

**for (int i = 0; i < 2; i++)**

**printf("Second i = %d\n", i);**

**}**

**}**