**מעבדה 2 - מת"מ**

**מגישים: דנאור סיני 311156616סהר עוז 311285514**

**תרגיל 1:**

#include<stdio.h>

#include<conio.h>

#include <math.h>

struct point //setting point sturct

{

float x;

float y;

}typedef point;

struct circle //setting circle struct

{

point center; //center is variable with point type

float radius;

}typedef circle;

int check\_circle(point \*p, circle \*cir);

void main()

{

point p;

circle cir;

printf("Enter the coordinates of your point: ");

scanf("%f %f", &p.x, &p.y);

printf("Enter the radius and the center of your circle: ");

scanf("%f %f %f", &cir.radius, &cir.center.x, &cir.center.y);

if (check\_circle(&p,&cir))//if check\_circle returned 1 the point is inside the circle

{

printf("\nThe point is included in the circle");

}

else//if isinscircle returned 0 the point is not inside the circle

{

printf("\nThe point is not in the circle");

}

getch();

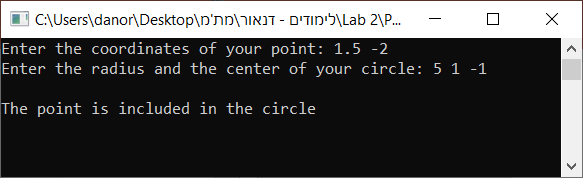
}

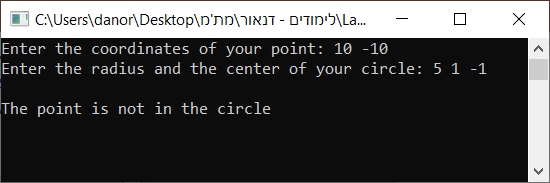
int check\_circle(point \*p, circle \*cir)/\*/using the circle formula (x-x0)+(y-y=)=r^2 to check if the point is inside /\*/

{

return (pow(cir->center.x - p->x, 2) + pow(cir->center.y - p->y, 2) <= pow(cir->radius, 2));

}





**תרגיל 2:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define MAX\_LEN 50

typedef struct stud

{

char \*name;

int marks[4];

float avg;

}student;

student\* Create\_Class(int);

void Avg\_Mark(student\*);

void Print\_One(student\*);

int main()

{

int size, i;

student \*arr;

printf("Enter the number of students:");

scanf("%d", &size);

arr = Create\_Class(size);//arr will point to the first element of the structs array that Create\_Class returns

for (i = 0; i < size; i++)

if (arr[i].avg > 85) Print\_One(arr + i); //printing only students with avarage higher than 85

for (i = 0; i < size; i++) //free every memory allocation of all names one by one

free(arr[i].name);

free(arr); //freeing the allocation of the structs array

getch();

return 0;

}

student\* Create\_Class(int size)//Create\_Class recieves the size of the class, and returns the address of the first element of the class array. it also sets all the information

{

student \*arr;

int i, j;

char str[MAX\_LEN];

arr = (student\*)malloc(size \* sizeof(student)); //Allocating memory for array of structs

if (arr == NULL) //if the allocation failed, print that there is not enought memory and exit the program

{

printf("Not enough memory");

exit(1);

}

for (i = 0; i < size; i++) //Setting info for each student

{

printf("Enter your name: ");

scanf("%s", str);

arr[i].name = (char\*)malloc((strlen(str) + 1) \* sizeof(char)); //allocating memory for the name string. (+1 for '\0')

if (arr[i].name == NULL) //if allocation failed, print an error and exit the program

{

printf("Not enough memory");

free(arr); //before exiting, freeing the allocation of the array

exit(1);

}

strcpy(arr[i].name, str); //copying the string from str to the allocated string

printf("Enter your marks:");

for (j = 0; j < 4; j++)

scanf("%d", &arr[i].marks[j]);

Avg\_Mark(arr + i); //calling Avg\_Mark for each student

}

return arr;

}

void Avg\_Mark(student\* s)//The function recieves address of student struct and sets variable avg in it

{

int i;

float avg = 0;

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

avg = avg + s->marks[i];

s->avg = avg / 4;

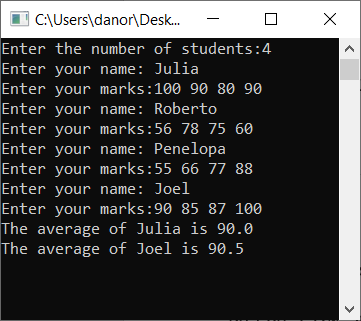
}

void Print\_One(student\* s)//the function recieves address of students and prints its name and average

{

printf("The average of %s is %.1f\n", s->name, s->avg);

}



**תרגיל 3:**

#include<stdio.h>

#include<conio.h>

#include<math.h>

#define NUM 3

struct complexnum

{

float re;

float im;

}typedef complexnum;

float CRadius(complexnum\* num);

complexnum\* Cmax(complexnum\* arr);

int main()

{

int i;

complexnum \*max;

complexnum arr[NUM];

printf("Enter 3 complex numbers:\n");

for (i = 0; i < NUM; i++)

scanf("%f %f", &arr[i].re, &arr[i].im);//the user will enter a complex number

max = Cmax(arr);

printf("The max complex number is %.2f+%.2fi", max->re, max->im);//will print the result

printf("\nThe radius if the max number is %.2f", CRadius(max));//will print the result

getch();

}

float CRadius(complexnum\* num)

{

return pow((num->re\*num->re) + (num->im\*num->im), 0.5);//the fucntion will calculate the radius of the complex number

}

complexnum\* Cmax(complexnum\* arr)

{

int i, max = 0;

for (i = 1; i < NUM; i++)

{

if (CRadius(arr + max) < CRadius(arr + i)) max = i;//the function will check which of the 3 complex numbers have the biggest radius

}

return (arr + max);

}

