**דוח מעבדה תרגיל בית 1**

**שמות מגישים:**

סער ויקטור – 312392822

אילון בן סימון – 312162951

**תרגיל 1**

קוד התכנית:

#include <stdio.h>

#include <math.h>

#include <conio.h>

#define N 10

//define structure

typedef struct Fraction

{

int mone, mehane;

}Fraction;

//functions decleration

void fractionScanAndBuild(Fraction \*arr);

int mehaneCommon(int a, int b);

int mone(Fraction \*arr, int c);

void negToPos(Fraction \*arr);

int main()

{

int i;

int newMehane, newMone;

Fraction arr[N];

printf("Enter fractions(max 9) by pairs of integers(0 0 to finish):\n");

fractionScanAndBuild(arr);

negToPos(arr); //converting negate mone and mehane to positive

newMehane = mehaneCommon(arr[1].mehane, arr[2].mehane); //initial common mehane

for (i = 1; i < N; i++) //calculating the mehane of all fractions

{

if (arr[i].mehane == 0)

break;

newMehane = mehaneCommon(newMehane, arr[i].mehane);

}

newMone = mone(arr, newMehane); //calculating new common mone

int x = newMehane;

int j = newMone%newMehane;

for (i = 2; i < N; i++) //loop to make the remainder the most reduced

{

if (newMehane%i == 0 && j%i == 0)

{

newMehane /= i;

j /= i;

i = 2;

}

}

printf("%d/%d", arr[1].mone, arr[1].mehane);

for (i = 1; i < N-1; i++) //loop for printing the end result given the sign of each fraction

{

if (arr[i + 1].mone == 0 && arr[i + 1].mehane == 0)

break;

if (arr[i + 1].mone>0)

printf("+%d/%d", arr[i + 1].mone, arr[i + 1].mehane);

else

printf("%d/%d", arr[i + 1].mone, arr[i + 1].mehane);

} //end loop

if (newMone/x < 0 && j < 0)

j \*= -1;

printf(" = %d/%d = %d and %d/%d", newMone, x, (newMone / x), j, newMehane);

getch();

return 0;

}

//the function builds the array of structures correctly

void fractionScanAndBuild(Fraction \*arr)

{

int i;

for (i = 1; i < N; i++) //scanning all fractions as long as both mone and mehane aren't zero

{

printf("%d (mone mehane) : ", i);

scanf("%d %d", &arr[i].mone, &arr[i].mehane);

if (arr[1].mone == 0 && arr[1].mehane == 0) //end program if the only numbers insterted are 0 and 0

{

printf("Bye Bye.");

getch();

exit(0);

}

while (arr[i].mone != 0 && arr[i].mehane == 0) //asking for a valid fraction if the mehane is 0

{

printf("Please enter valid numbers (mehane != 0):\n");

scanf("%d %d", &arr[i].mone, &arr[i].mehane);

if (arr[1].mone == 0 && arr[1].mehane == 0) //end program if the only numbers insterted are 0 and 0

{

printf("Bye Bye.");

getch();

exit(0);

}

}

if (arr[i].mone == 0 && arr[i].mehane == 0)

break;

}

}

//the function returns the common mehane of 2 fractions

int mehaneCommon(int a, int b)

{

int mehaneCommon;

if (a > b)

mehaneCommon = a;

else

mehaneCommon = b;

while (1)

{

if (b == 0)

break;

if (mehaneCommon%a == 0 && mehaneCommon%b == 0)

return mehaneCommon;

mehaneCommon++;

}

}

//the function returns the common mone of all fractions

int mone(Fraction\*arr, int c)

{

int i, new\_mone = 0;

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

{

if (arr[i].mehane == 0)

break;

new\_mone += (c\*arr[i].mone) / arr[i].mehane;

}

return new\_mone;

}

void negToPos(Fraction \*arr)

{

int i;

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

{

if (arr[i].mone < 0 && arr[i].mehane < 0) //if both mone and mehane are negative, turns the fraction to positive

{

arr[i].mone \*= -1;

arr[i].mehane \*= -1;

}

else if (arr[i].mone > 0 && arr[i].mehane < 0) //if the mone is positive and the mehane is negative, switches signs between them

{

arr[i].mone \*= -1;

arr[i].mehane \*= -1;

}

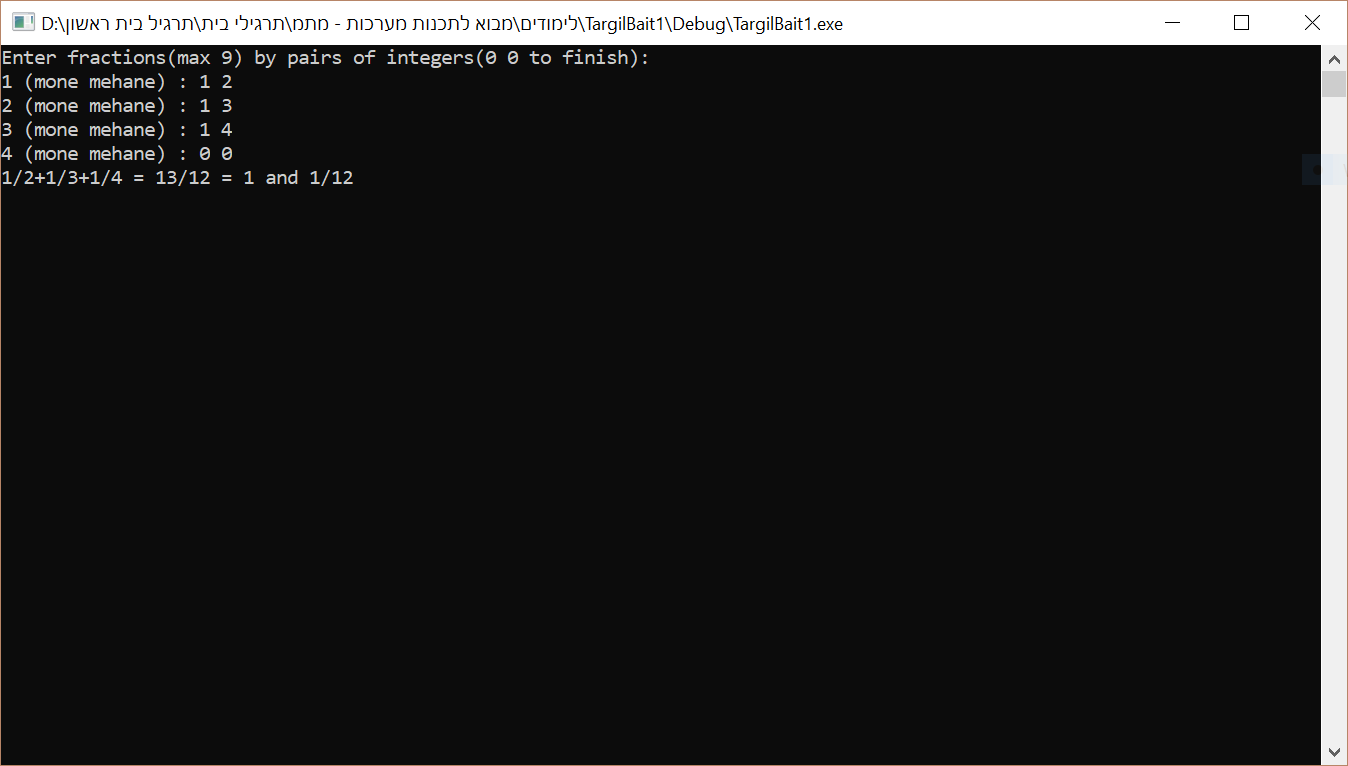
if (arr[i].mone == 0 && arr[i].mehane == 0)

break;

}

}

פלט לדוגמא:



**תרגיל 2**

קוד התכנית:

#include <stdio.h>

#include <conio.h>

#include <math.h>

#include <stdlib.h>

#include <string.h>

#define N 5

//define structure

typedef struct Division

{

char code[10];

char \*name;

int numOfProducts;

int soldProductsToday;

}Division;

Division\* build();

int main()

{

int i;

Division\* pDivision = build();

Division \*temp, \*temp1;

Division \*\*pArr;

pArr = (Division\*\*)malloc(sizeof(Division\*)\*N); //memory allocation for the array of pointers

if (pArr == NULL)

{

printf("Error! Memory Not Allocated!");

exit(0);

}

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

{

pArr[i] = &pDivision[i]; //matching the indexes of the pointer's array to the structure's array

}

int max, i\_max, min, i\_min;

max = pDivision[0].soldProductsToday; //initializing the max to the first index

min = pDivision[N - 1].soldProductsToday; //initializing the min to the last index

i\_max = 0;

i\_min = N - 1;

for (i = 0; i < N; i++) //start loop for finding the max, max index, min and min index

{

if (pDivision[i].soldProductsToday>max)

{

max = pDivision[i].soldProductsToday;

i\_max = i;

}

if (pDivision[i].soldProductsToday < min)

{

min = pDivision[i].soldProductsToday;

i\_min = i;

}

} //end loop

temp = pArr[0];

temp1 = pArr[N - 1];

pArr[0] = &pDivision[i\_max];

pArr[N - 1] = &pDivision[i\_min];

for (i = 1; i < N - 1; i++) //start loop for checking if an adress shows up twice

{

if (&pDivision[i\_max] == pArr[i])

{

pArr[i] = temp;

break;

}

} //end loop

for (i = 1; i < N - 1; i++) //start loop for checking if an adress shows up twice

{

if (&pDivision[i\_min] == pArr[i])

{

pArr[i] = temp1;

break;

}

} //end loop

printf("\nBest Division:\n");

printf("Name: %s\nCode: %s", pArr[0]->name, pArr[0]->code);

printf("\nWorst Division:\n");

printf("Name: %s\nCode: %s", pArr[N - 1]->name, pArr[N - 1]->code);

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

{

free(pDivision[i].name);

}

free(pDivision);

free(pArr);

getch();

return 0;

}

Division\* build()

{

char temp[20];

int i;

Division \*pDivision;

pDivision = (Division\*)malloc(sizeof(Division)\*N); //memory allocating for the structure's array

if (pDivision == NULL)

{

printf("Error! Memory Not Allocated!");

exit(0);

}

printf("Enter %d division's stats:", N);

for (i = 0; i < N; i++) //start loop for scanning all the stats

{

printf("\nDivision's code (10 chars): ");

scanf("%s", pDivision[i].code); //code scanning

printf("Division's name (maximum 20 chars): ");

scanf("%s", temp); //name scanning

pDivision[i].name = (char\*)malloc(strlen(temp) + 1); //memory allocating for each division's name

if (pDivision[i].name == NULL)

{

printf("Error! Memory Not Allocated!");

exit(0);

}

strcpy(pDivision[i].name, temp); //copying the temp string to the name

while (1) //scan valid stats

{

printf("Number of products in the division: ");

scanf("%d", &(pDivision[i].numOfProducts)); //num of products scanning

printf("Number of products sold per day: ");

scanf("%d", &(pDivision[i].soldProductsToday)); //sold products today scanning

if (pDivision[i].soldProductsToday < pDivision[i].numOfProducts)

break;

else

printf("Please enter valid stats:\n");

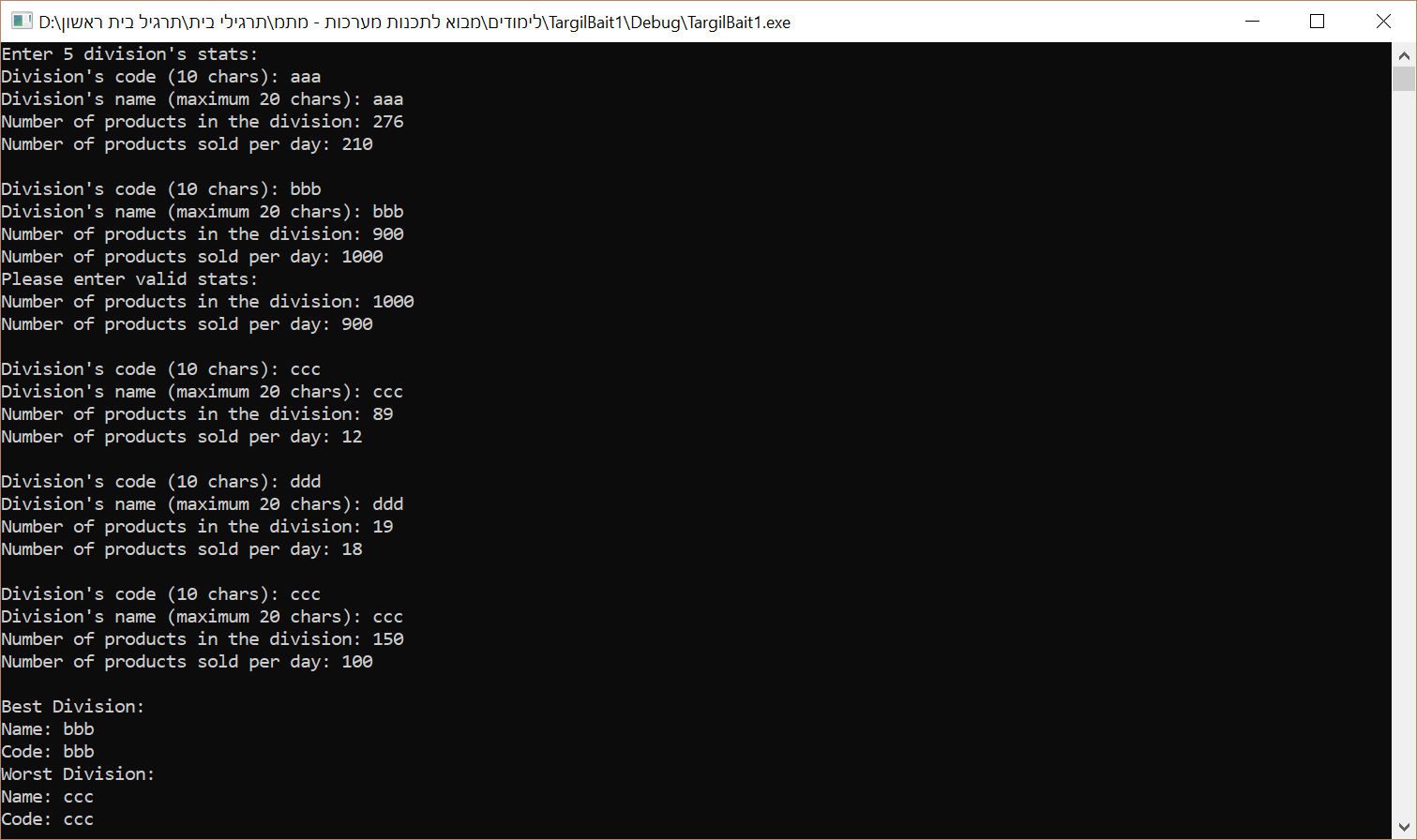
}

} //end loop

return pDivision;

}

פלט לדוגמא:



**תרגיל 3**

קוד התכנית:

#include <stdio.h>

#include <math.h>

#include <conio.h>

//define structure

typedef struct Clock

{

int hour, minute;

}clock\_t;

int main()

{

int addition;

clock\_t time;

int addMinutes;

printf("Enter time and minutes to add: ");

while (1) //scan a valid time

{

scanf("%d %d %d", &(time.hour), &(time.minute), &addMinutes);

if ((time.hour < 25) && (time.hour>=0) && (time.minute < 60) && (time.minute >= 0))

break;

else

printf("Please enter a valid time:\n");

}

if (time.hour == 24)

time.hour -= 24;

if (time.hour >= 0 && time.hour <= 9)

{

if ((time.minute >= 0) && (time.minute <= 9))

printf("The time is: 0%d:0%d\n", time.hour, time.minute);

else

printf("The time is: 0%d:%d\n", time.hour, time.minute);

}

else

{

if ((time.minute >= 0) && (time.minute <= 9))

printf("The time is: %d:0%d\n", time.hour, time.minute);

else

printf("The time is: %d:%d\n", time.hour, time.minute);

}

addition = time.minute + addMinutes; //calculates the total minutes

if (addition < 60)

time.minute = addition;

else

while (addition >= 60) //the loop calculates the minutes and hours to add

{

time.hour += 1;

time.minute = addition - 60;

addition -= 60;

if (time.hour == 24)

time.hour -= 24;

}

if (time.hour >= 0 && time.hour <= 9)

{

if ((time.minute >= 0) && (time.minute <= 9))

printf("The new time is: 0%d:0%d", time.hour, time.minute);

else

printf("The new time is: 0%d:%d", time.hour, time.minute);

}

else

{

if ((time.minute >= 0) && (time.minute <= 9))

printf("The new time is: %d:0%d", time.hour, time.minute);

else

printf("The new time is: %d:%d", time.hour, time.minute);

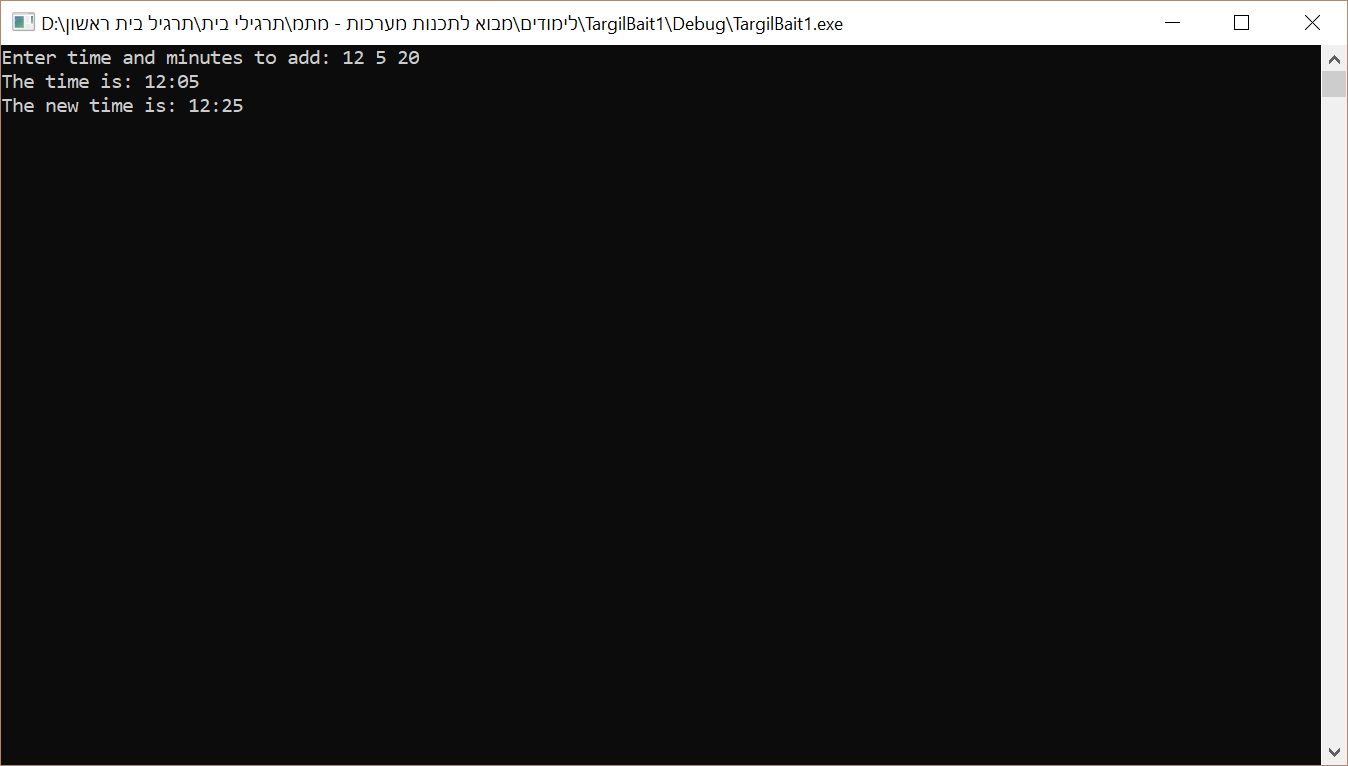
}

getch();

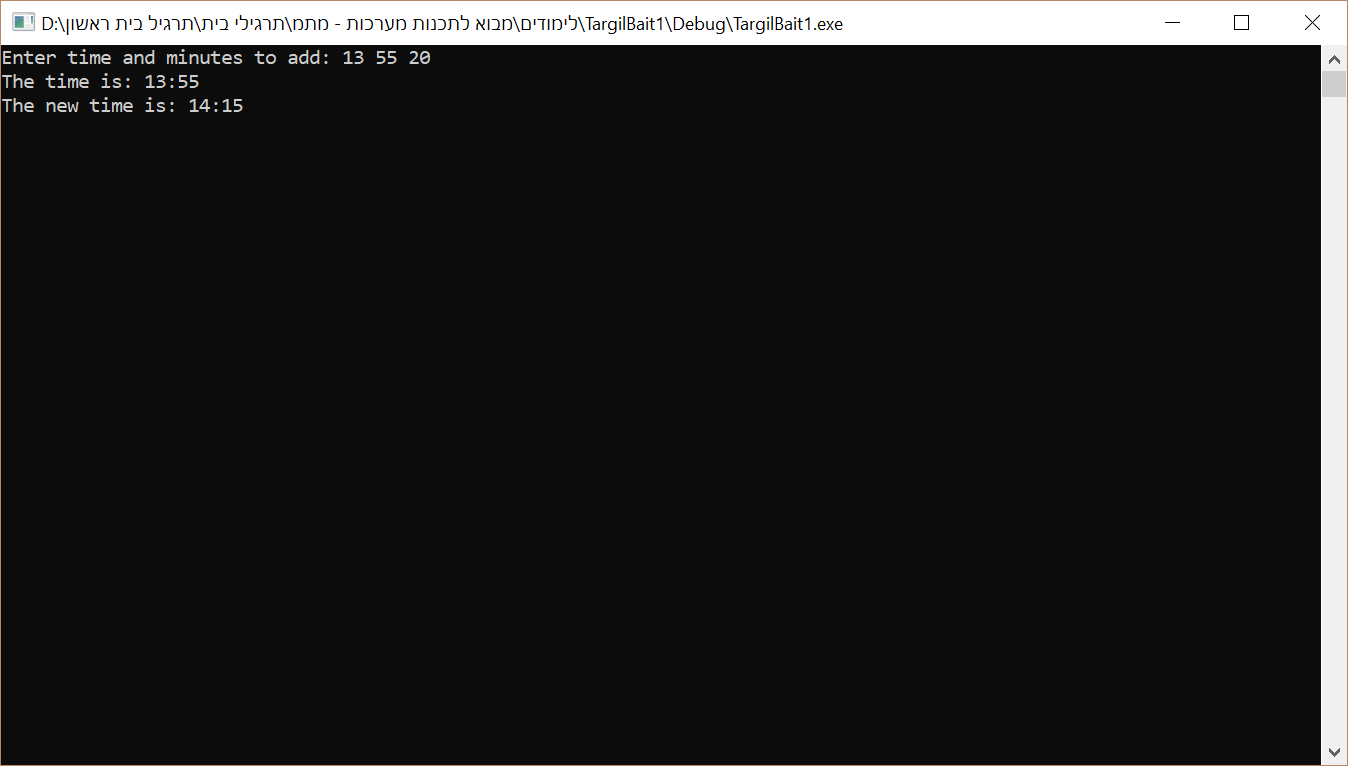
return 0;

}

פלט 1 לדוגמא:



פלט 2 לדוגמא:



פלט 3 לדוגמא:

