**תוכנית תרגיל 2 שאלה 1**

// Foreground/Background Sample Program

// ==================================

/\*

This program demonstrates a foreground program (system timer interrupt

routine) that is invoked every 45 milliseconds and a background program

that scans an 18 ticks counter and displays a seconds counter every

second.

\*/

#include <dos.h>

#include <conio.h>

#include <iostream.h>

#include <stdlib.h>

#define timer 8 // The timer interrupt routine entry number

void interrupt(\*oldhandler)(...); // old ISR save memory

int hours, minutes, seconds; // time of the day;

int count18 = 0; // ticks counter(18 ticks are approximately 1 sec)

int flag = 0; // 1 second expiration flag

//our params

char\* z[2] = { "AM", "PM" };

int index = 0;

bool isAm;

int totSec = 0;

int X; //timeout param

bool flag2 = true;//chooses between kbhit and timeout according to user input

//

//-----------------------------------

// Timer Interrupt Routine

//-----------------------------------

void interrupt handler(...) // Define timer interrupt routine.

{

count18++; // Increment the ticks counter

if (count18 >= 18) // if 1 second expired

{

count18 = 0; // re - init ticks counter

flag = 1; // indicate 1 second expiration

}

oldhandler(); // return via operating system ISR

}

//-----------------------------------

//Initialization function

//-----------------------------------

void initiate()

{

start: clrscr(); // clear screen

gotoxy(20, 5);

// Read current time of the day

cout << " enter current time hh mm ss (24 hours) :";

cin >> hours >> minutes >> seconds;

// if input is not valid - return with a message...

if (hours < 1 || hours >24

|| minutes < 0 || minutes >59

|| seconds < 0 || seconds >59)

// exit(0);

{

gotoxy(28, 10);

cout << "Wrong time - enter new one again";

delay(5000);

goto start;

}

//after time params are checked infer am/pm and fix params

if (hours < 24 && 11 < hours)

{

if (hours != 12) { hours = hours - 12; }

index = 1;

isAm = false;

}

else

{

if (hours == 24) { hours = hours - 12; }

index = 0;

isAm = true;

}

//get timeout param from user

start2:

cout << " Please Enter Timeout in sec (999 is anykey):";

cin >> X;

if (X < 0 || 999 < X)

{

gotoxy(28, 10);

cout << "Wrong X/timeout - try again ";

delay(5000);

clrscr(); // clear screen

goto start2;

}

clrscr(); // clear screen

gotoxy(28, 1);

cout << "System Timer Test Program ";

if (X == 999)

{

gotoxy(28, 10);

cout << "The time is:";

gotoxy(20, 20);

cout << "Press any key to stop";

}

else if (X == 0)

{

gotoxy(28, 10);

cout << ":)";

gotoxy(20, 20);

cout << "time out is zero";

}

else

{

gotoxy(28, 10);

cout << "The time is:";

gotoxy(20, 20);

cout << "time out is set";

}

// critical section -

disable(); // disable interrupts

oldhandler = getvect(timer); // Save the old interrupt vector

setvect(timer, handler); // Install the new interrupt handler

enable(); // enable interrupts

}

//--------------------------------

// Termination function

//--------------------------------

void terminate()

{

//clrscr(); // clear screen

gotoxy(34, 15);

cout << "Program terminated !"; // End message

// critical section

disable(); // disable interrupts

setvect(timer, oldhandler); // reload operating system ISR to IVT

enable(); // enable interrupts

}

//-----------------------------------------------------------------

//Main Program (Initialization and Background)

//-----------------------------------------------------------------

void main()

{

initiate(); // Initialize peripherals , inits with user am/pm status

while (flag2)

{

//chooses between keyboard hit and timeout

if (X == 999)

{

flag2 = !kbhit();

}

else

{

if (totSec >= X)

{

flag2 = false;

flag = 0;

}

}

if (flag == 1) // If 1 second passed

{

// critical section

disable(); // disable interrupts

flag = 0; // zero second expiration flag

enable(); // enable interrupts

seconds = seconds + 1; // increment seconds count

// check for end of a minute

if (seconds >= 60)

{

seconds = 0; // zero seconds counter

minutes = minutes + 1; // increment minutes count

// check for end of an hour

if (minutes >= 60)

{

// zero minutes counter

minutes = 0; // increment hours count

hours = hours + 1;

if (hours > 12)

{

hours = 1; // if 12 hours passed - hour is 1 again

}

if (hours == 12) //make AM/PM periodic

{

if (isAm)

{

isAm = false;

index = 1;

}

else

{

isAm = true;

index = 0;

}

} //hours

}

}//if sec>=60

gotoxy(44, 10);

cout << " "; // clear tod area

gotoxy(44, 10);

cout << hours << " : " << minutes << " : " << seconds << " " << z[index]; // display tod

//count seconds for timeout and running time

totSec = totSec + 1;

}//if flag==1

}//while

//gotoxy(44,10);

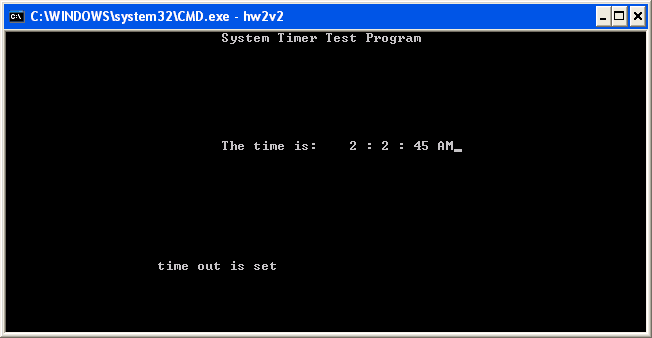
cout << "\n" << "runtime (in sec) : " << totSec << endl;

// If any key hit - terminate the program

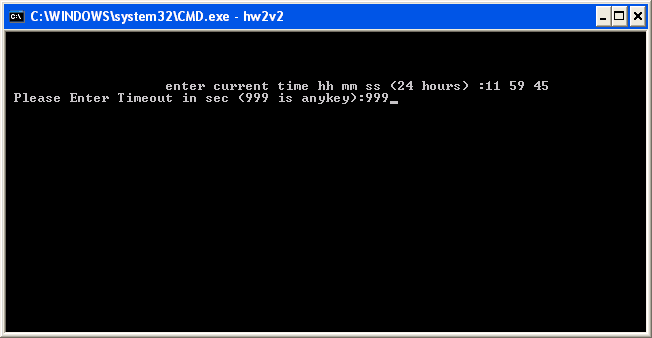
terminate(); // restore the old interrupt handlers and I/O registers & exit

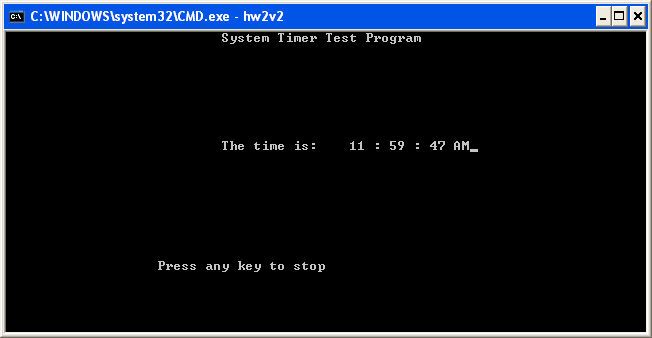
}

**צילומי מסך תרגיל 2 שאלה 1**

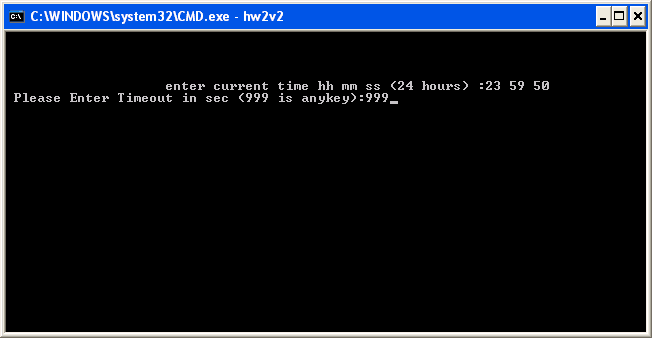


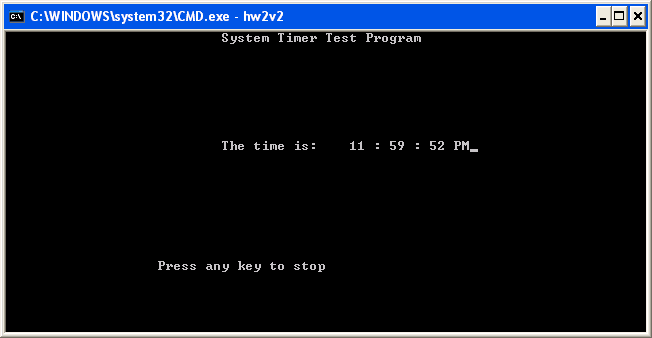
תצוגת זמן תקינה



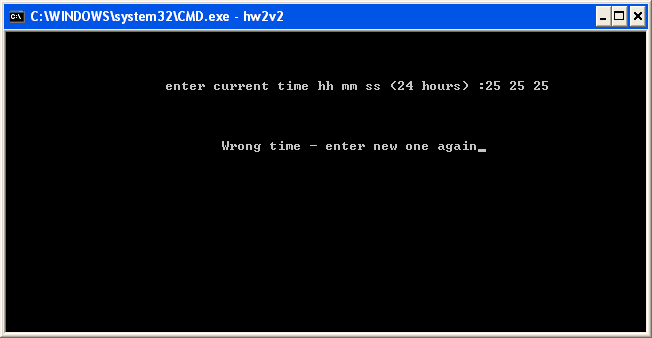


תצוגת AM תקינה

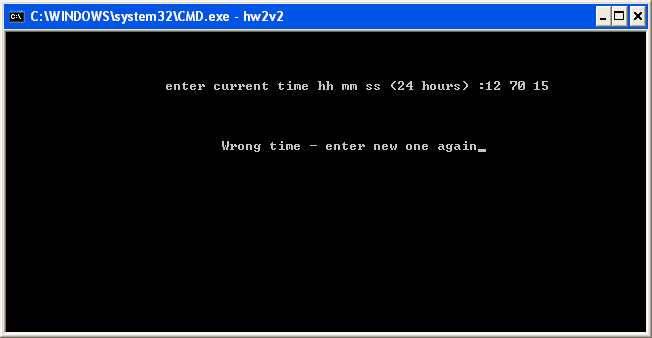




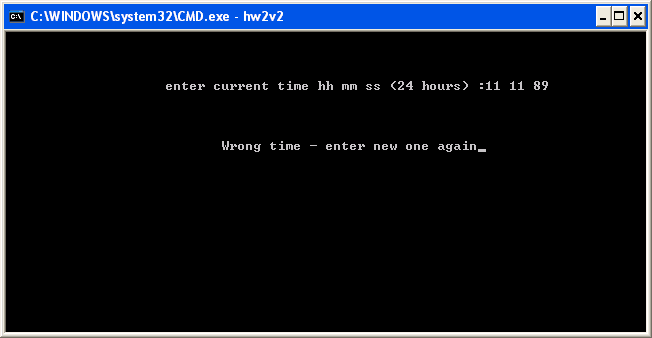
מעבר ל PM תקין



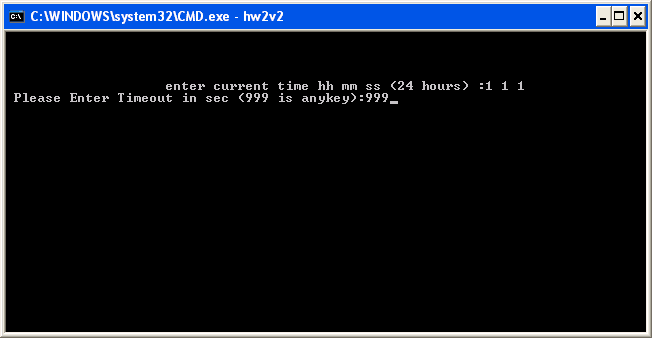
בדיקת שעות (קלט)

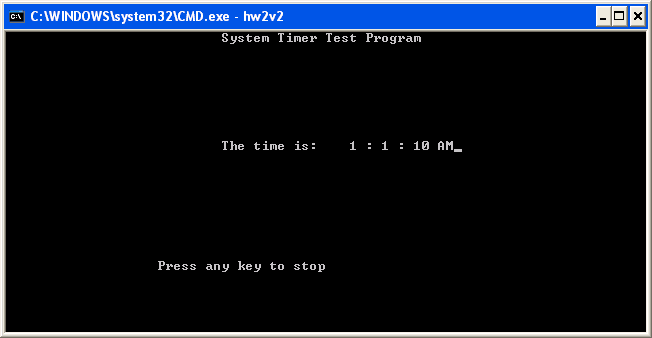


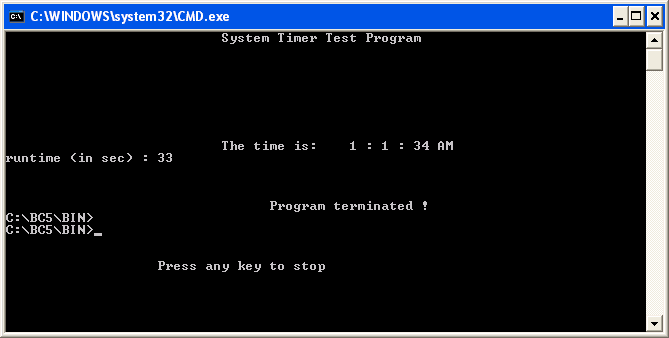
בדיקת דקות (קלט)



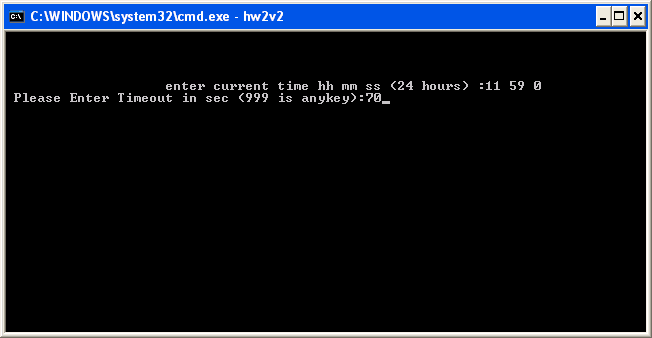
בדיקת שניות (קלט)

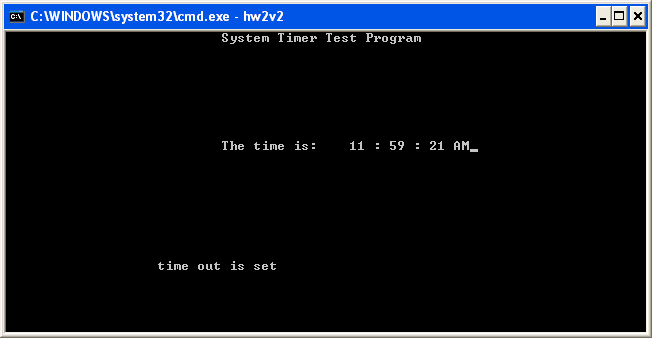


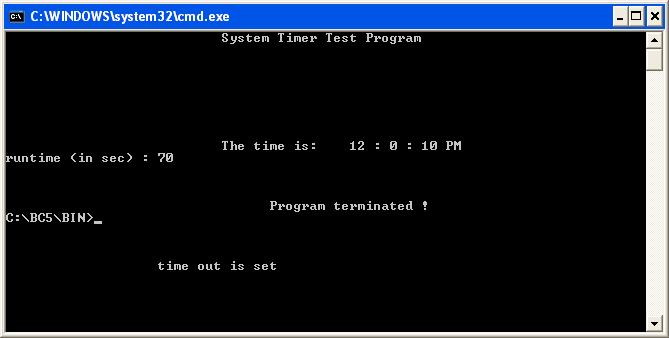




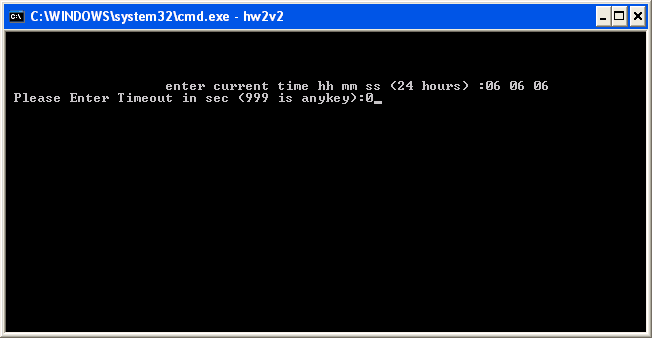
שלושת המסכים האחרונים מתייחסים לבדיקת קלט 999 (עצירה על ידי כל מקש)

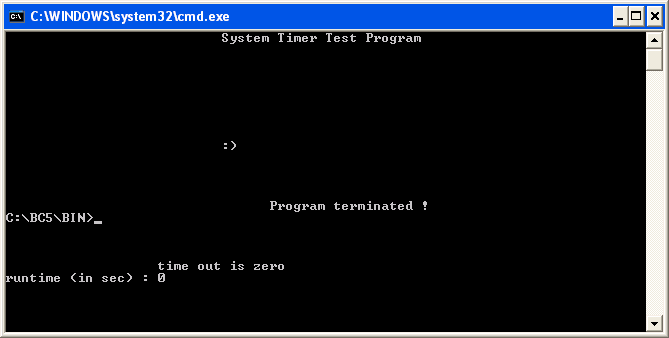






בדיקת טיימאאוט (70 שניות)





בדיקת טיימאאוט (0 שניות)