ASSEMBLLY CODE:

```
.MODEL SMALL
.STACK 100H
.DATA
    MSGA DB 'Hashmats Soldier: $'
    MSGB DB 'Opponents Soldier: $'
    MSGC DB 'Result: $'
    MSGD DB 13,10,'$'
    COUNT DB ?
    HS DW ?
    OS DW ?
    RESULT DW ?
    TEMP DW ?

.CODE
MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

RAND_IO:
    MOV COUNT,2
    LEA DX,MSGA                 ;Print Hashmat soldier'
    MOV AH,9
    INT 21H

    GET_INPUTS:
        MOV DX,0
        MOV BX,0                ;Clear bx
        MOV AH,1
        INT 21H

        CMP AL,0DH
        JE END_INPUTS           ;If enter

        CONVERT_TO_NUM:
            AND AX,000FH        ;Use full 16 bits of AX
            MOV TEMP,AX
            MOV AX,10
            MUL BX
            MOV BX, AX
            ADD BX,TEMP

        MOV AH,1                ;Input new digit
        INT 21H
        CMP AL,0DH
        JNE CONVERT_TO_NUM

        END_INPUTS:
            CMP COUNT,1
            JNE GET_HS

            MOV OS,BX           ;Get Time
            JMP GET_OS

            GET_HS:
                MOV HS,BX       ;Get Hasmat Solders
                LEA DX,MSGB     ;Print 'Opponents Soldier:'
                MOV AH,9
                INT 21H

            GET_OS:
                DEC COUNT
                CMP COUNT,0
                JNE GET_INPUTS
```

```
    GET_RESULT:
        MOV BX,HS
        MOV AX,OS
        CMP BX,OS
        JGE FIRST_CON
        JMP SECOND_CON

        FIRST_CON:
            SUB BX,OS
            MOV RESULT,BX
            JMP START_PRINT

        SECOND_CON:
            SUB AX,HS
            MOV RESULT,AX

    START_PRINT:
        LEA DX,MSGC             ;Print result
        MOV AH,9
        INT 21H

        MOV AX,RESULT
        MOV CX,0
        MOV BX,10

        STOR_RESULTS:           ;stor each digits in stack
            XOR DX,DX
            DIV BX
            PUSH DX
            INC CX
            CMP AX,0
            JNE STOR_RESULTS

        PRINT_RESULTS:          ;print each digits from stack
            MOV AH,2
            POP DX
            ADD DL,48
            INT 21H
            LOOP PRINT_RESULTS

    NEW_INPUT:
        LEA DX,MSGD
        MOV AH,9
        INT 21H
        JMP RAND_IO

MAIN ENDP
    END MAIN
```