**RAJSHAHI UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

**Course No:** CSE 3110

**Course Title:** Sessional Based on CSE 3109

**Problem No:** 05

**Problem Name:** Write a program that lets the user enter time in seconds, up to 65535 and outputs the time as hours, minutes and seconds

**Submitted To:**

Sadia Zaman Mishu

Assistant Professor

Computer Science and Engineering

Rajshahi University of Engineering and Technology

**Submitted By:**

Name : Md. Fakhrul Islam

Roll: 1803071

Session: 2018-19

Department : Computer Science and Engineering

Rajshahi University of Engineering and Technology

**Problem No:** 05

**Problem Name:** Write a program that lets the user enter time in seconds, up to 65535 and outputs the time as hours, minutes and seconds

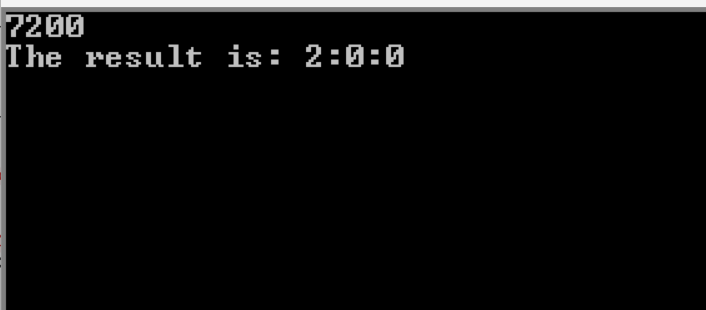
**Description:**

There are separate instructions for signed and unsigned division. For unsigned division, DIV instruction is used. IDIV is used for signed division. For 8 bits divisor, AX is the dividend. After performing division operation, the remainder is in AL and the quotient is in AH. For 16-bit divisor, the 32-bit dividend is assumed to be in DX:AX. In this case, the remainder is in AX and the quotient is in DX. If the divisor is much smaller than the dividend, the divide overflow will occur.

**Code:**

|  |  |
| --- | --- |
| .MODEL SMALL  .STACK 100H  .DATA  MSG DB 0AH,0DH,'The result is : $'  .CODE  MAIN PROC  MOV AX,@DATA  MOV DS,AX  CALL INDEC  MOV DX,0  MOV BX,3600  DIV BX  PUSH AX  PUSH DX  MOV AH,9  LEA DX,MSG  INT 21H  POP DX  POP AX  CALL OUTDEC  PUSH DX  MOV AH,2  MOV DL,':'  INT 21H  POP AX  MOV BX,60  MOV DX,0  DIV BX  CALL OUTDEC  PUSH DX  MOV AH,2  MOV DL,':'  INT 21H  POP AX  CALL OUTDEC  MOV AH,4CH  INT 21H  MAIN ENDP  INDEC PROC  PUSH BX  PUSH CX  PUSH DX  BEGIN:  XOR BX,BX  XOR CX,CX  MOV AH,1  INT 21H  CMP AL,'-'  JE MINUS  CMP AL,'+'  JE PLUS  JMP REPEAT2  MINUS:  MOV CX,1  PLUS:  INT 21H  REPEAT2:  CMP AL,'0'  JNGE NOT\_DIGIT | CMP AL,'9'  JNLE NOT\_DIGIT  AND AX,000FH  PUSH AX  MOV AX,10  MUL BX  POP BX  ADD BX,AX  MOV AH,1  INT 21H  CMP AL,0DH  JNE REPEAT2  MOV AX,BX  OR CX,CX  JE EXIT  NEG AX  EXIT:  POP DX  POP CX  POP BX  RET  NOT\_DIGIT:  MOV AH,2  MOV DL,0DH  INT 21H  MOV DL,0AH  INT 21H  JMP BEGIN  INDEC ENDP  OUTDEC PROC  PUSH AX  PUSH BX  PUSH CX  PUSH DX  OR AX,AX  JGE END\_IF1  PUSH AX  MOV DL,'-'  MOV AH,2  INT 21H  POP AX  NEG AX  END\_IF1:  XOR CX,CX  MOV BX,10D  REPEAT1:  XOR DX,DX  DIV BX  PUSH DX  INC CX  OR AX,AX  JNE REPEAT1  MOV AH,2  PRINT:  POP DX  OR DL,30H  INT 21H  LOOP PRINT  POP DX  POP CX  POP BX  POP AX  RET  OUTDEC ENDP  END MAIN |

**Output:**



Conclusion:

For input and output, INDEC and OUTDEC procedure is used. If the divisor is 8 bit and the dividend is also 8 bit, the AH should be cleared. Similarly, if the divisor and dividend are 16 bit, the DX should be cleared for unsigned division. For signed division, DX will be the sign extension of AX.

In this program, if the input is a non-digit, the program will skip the line and take a new input. But for a negative number, it will display a wrong answer. We can solve this problem just like non-digit input.