// DATE 22/04/2015 | AUTHOR-SUNNY SAINI ,YOGESH SAINI ,PANKAJ KUMAR

// NAME OF PROJECT -ALARM CLOCK | SUBJECT-SOFTWARE ENGINEERING

#include<stdio.h> // header file to manage input output function

#include<time.h> // header file to take the processor time

#include<math.h> // header file for including mathematics functions

#include<windows.h> // header file to include the BEEP function for alarm

void split\_time(long int total\_sec,int \*hr,int \*min, int \*sec); // declaring split function for spliting total seconds into HOUR MIN SEC

void customize\_time(int \*,int\*,int\*); // function to customize time if user enter more than 24 hour or 60 min

void set\_alarm(int,int); // function to write alarm time in data file

void manage\_alarm(int \*,int \*); // function to read input from user for alarm

int main() // begin of main function

{

long y,s; // calculate passed year and seconds

char ch;

int hr,min,sec,alarm\_hour,alarm\_min,i=100000;

clock\_t ts;

printf("\t\t\t CURRENT TIME HH :MM :SS\n");

while(i)

{

ts=time(NULL);

y=ts%(365\*24\*60\*60);

s=y%(24\*60\*60);

split\_time(s,&hr,&min,&sec);

customize\_time(&hr,&min,&sec);

printf("\t\t\t\t\t%d :%d :%d \r",hr,min,sec);

i--; // print processor time

}

printf("\n");

manage\_alarm(&alarm\_hour,&alarm\_min); // taking a variable to store total second from 1st jan 1970 to current time

FILE \*fp; // a file pointer

fp=fopen("alarm.txt","r"); // opening alarm.txt file in reading mode

fscanf(fp,"%d%c%d",&alarm\_hour,&ch,&alarm\_min); // take input from file for alarm

fclose(fp); // closing file

if(alarm\_hour>=24||alarm\_min>=60) // if alarm time is not in proper format then take action about it

manage\_alarm(&alarm\_hour,&alarm\_min); // if user enter wrong data then ask for re-input

while(1)

{

ts=time(NULL);

y=ts%(365\*24\*60\*60);

s=y%(24\*60\*60);

split\_time(s,&hr,&min,&sec);

customize\_time(&hr,&min,&sec);

if(hr==alarm\_hour&&min==alarm\_min) // if processor time equal to alarm time the do operation for BEEP

{

Beep(5000,600); // call beep function to produce beep

set\_alarm(24,0); //

}

printf("\t\t\t\t\t%d :%d :%d \r",hr,min,sec); // print processor time

}

return 0; // return NULL after successfull runing of program

} // ending of main function

void split\_time(long int total\_sec,int \*hr,int \*min,int \*sec) // defining split function to split total seconds

{

int h;

\*hr=total\_sec/(60\*60); // calculating hour fron seconds

h=total\_sec%(60\*60); // remain seconds

\*min=h/60; // calculating min from remain seconds

\*sec=h%60; // seconds

}

void customize\_time(int \*hour ,int\*minute ,int\* second) // customize function to customize hour min and seconds

{ // that is calculated from the function SPLIT\_TIME

\*hour=\*hour+5; // like if MIN is greater then 65 min then increase HOURS

\*minute=\*minute+30; // and re-claculate MIN

if(\*minute>59)

{

\*hour=\*hour+1;

\*minute=\*minute-60;

}

if(\*hour>=24)

\*hour=\*hour-24;

}

void set\_alarm( int hour,int minute) // set alarm data into file so that it can be use

{ // to sound alarm

FILE \*fp; // declare file pointer and write alarm data into file

fp=fopen("alarm.txt","w");

fprintf(fp,"%d:%d",hour,minute);

fclose(fp); // close file after writing data

}

void manage\_alarm(int \*hour,int \*minute) // define manage\_alarm function to read alarm data from user

{

int hh,mm; // if user enter wrong data then show warnning and request to re-enter

printf("\rEnter Alarm Time(24)(HH MM) : ");

scanf("%d%d",&hh,&mm);

if(hh>=24||mm>=60||hh<0||mm<0) // show warnning if invalid time is provided wrong

{

printf("enter valid time please ");

manage\_alarm(hour,minute); // it is a recursive function

} // it's run till user enter valid time

else

set\_alarm(hh,mm); // call set\_alarm function

\*hour=hh;

\*minute=mm;

}

// END OF PROGRAM

// copyright 2015 p.s.y. company limited

// NO PART OF THIS PROGRAM MAY BE REPRODUCED IN ANY FORM

// OR BY ANY MEANS ,WITHOUT THE PROIR WRITTEN PERMISSION OF AUTHOR