**ALGORITHM**

201133216 정유석

**PROBLEM 1**

#include<stdio.h>

#include<stdlib.h>

#define MAX\_TIME 100

typedef struct{ //Structure -- Schedules time of work and appointments

int start, end; //start = start time of work, end = end time of work

}appt;

char line[255]; //

int returntime(char\*,char\*);//Calculate the time (translate the hour to minutes)

appt snap[100]; //It is store the time of schedules start or end

void main(){

int time[MAX\_TIME]; //Store the each nap time

int day=0; //It is date

int num\_apt;//Number of schedules

int i,j,snap\_num,snap\_time; // i,j is loop value -- snap\_num, snap\_time is temporary value

while(scanf("%d", &num\_apt) == 1){

gets(line); //receive the schedules

if(num\_apt>1 && num\_apt<100){ //number of schedules is not over than 100 , less than 1

day++;//increase date

for(i=0;i<num\_apt;i++){ //receive the schedules time

gets(line);

if(line[0]-'0'!=1){//Schedules time is only 10:00 ~ 19:59

printf("Wrong appointments 2! \n");

exit(0);

}

//Calculate the time of schedules -- It convert to minutes

snap[i].start = returntime(line,line+3);

snap[i].end = returntime(line+6, line+9);

}

for(i=0; i<num\_apt-1; i++){

time[i] = snap[i+1].start - snap[i].end ;

//Array of time is store the each nap time

if(time[i] < 0){ //If wrong the time of schedules, exit the program

printf("Wrong appointments 1! \n");

exit(0); //So schedules is not same time

}

}

for(i=0; i<num\_apt-1; i++){ //Find the longest nap time

if(i==0){//Set the temporaray value

snap\_num = i;

snap\_time = time[i];

}

else{

if(time[i]>snap\_time){ //So if same the longest time than select preview time

snap\_num = i;

snap\_time = time[i];

}

}

}

printf("Day #%d: the longest nap starts at ", day);

printf("1%d:%02d and will last for %d hour %d minutes\n ",snap[snap\_num].end / 60, snap[snap\_num].end % 60, time[snap\_num]/60, time[snap\_num]%60);

}

//If s>100 or s<2 than exit the program

else {

printf("Wrong appointments 3! \n");

break;

}

}

}

int returntime(char \*hh, char \*mm){

return (hh[1] - '0') \* 60 + (mm[0] - '0') \* 10 + (mm[1] - '0');

}

