**<201433707 이형욱 알고리즘 sweetnap>**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include<stdlib.h>

#include <string.h>

void main()

{

int count = -1;

char str[200]; // saving line

char tempstr[200];

FILE \*fp; //file pointer instance

char \*res;

char filename[100]; //file name

printf("write file name : ");

gets\_s(filename);

fp = fopen(filename, "r"); //file open

int start = 0; //schedule starting point

int last = 0; //schedule finishing point

int j = -1; //time array information

char sleep1[2]; //each line save time ;hour

char sleep2[2]; //each line save time ;minute

char sleep3[2]; //each line save time ;hour

char sleep4[2]; //each line save time ;minute

//int sleep2[7][8];

int sweetnap[10][10]; //Variables that store each time

int calnap[7]; //Variable that stores the number of all occasions that can sleep

int day = 0;

int len = 0;

if (fp == NULL)

{

printf("ERROR.\n");

exit(0);

}

while (1)

{

res = fgets(str, sizeof(str), fp); //Read the file one line at a time.

if (res == NULL)

break;

printf("%s", str);

len = strlen(str);

//Save as many as the number of schedules in variable last

if (len == 2) {

last = start + atoi(str);

}

else if (len != 2) {

start++;

j++;

//Store each time in a variable named sleep

for (int i = 0; i < 2; i++) {

sleep1[i] = str[i];

}

for (int i = 0; i < 2; i++) {

sleep2[i] = str[i+3];

}

for (int i = 0; i < 2; i++) {

sleep3[i] = str[i + 6];

}

for (int i = 0; i < 2; i++) {

sleep4[i] = str[i + 9];

}

for (int i = 0; i < 1; i++) {

//Save the text by changing the letters to numbers in the sentence

sweetnap[j][i] = atoi(sleep1)\*60; sweetnap[j][i + 1] = atoi(sleep2);

sweetnap[j][i + 2] = atoi(sleep3)\*(60);

sweetnap[j][i + 3] = atoi(sleep4);

}

}

if (start == 0) {

continue;

}

//When the schedule is over;

//Calculate and save the time you can sleep naps between schedules

else if (last == start) {

int i = 0;

int max = 0;

int when = 0;

calnap[0] = (sweetnap[0][0] + sweetnap[0][1])-600;

max = calnap[0];

for ( i = 1; i < last; i++)

{

calnap[i] = (sweetnap[i][0] + sweetnap[i][1]) - (sweetnap[i-1][2] + sweetnap[i-1][3]);

}

calnap[last] = (18 \* 60) - (sweetnap[last-1][2] + sweetnap[last-1][3]);

//Get the most sleep time

for (int k = 1; k <=last; k++)

{

if (max < calnap[k]) {

max = calnap[k];

when = k;

}

}

day++;

//print: sweet nap

printf("\nDay # %d: the longest nap starts at %d : ",day,sweetnap[when-1][2]/60);

if (sweetnap[when - 1][3] == 0)

printf("00");

else

printf("%d", sweetnap[when - 1][3]);

printf("(finish scheduel) and will last for %d hours and %d minutes \n", max / 60, max % 60);

start = 0;

last = 0;

j = -1;

//Initializing

for (int y = 0; y < 7; y++)

{

calnap[y] = 0;

}

}

}

fclose(fp);

printf("\n");

}

**<print Screen>**

