**INFORME PARCIAL 1**

**Informática II**

**Juan Sebastian Marin Valencia**

**Despartamento de Ingeniería Electronica y**

**Telecomunicaciones**

**Universidad de Antioquia**

**Medellín**

**Índice**

**2.índice**

**3.Analisis general**

**5.funciones planeadas**

**ANALISIS PARCIAL 1**

**INFORMATICA II**

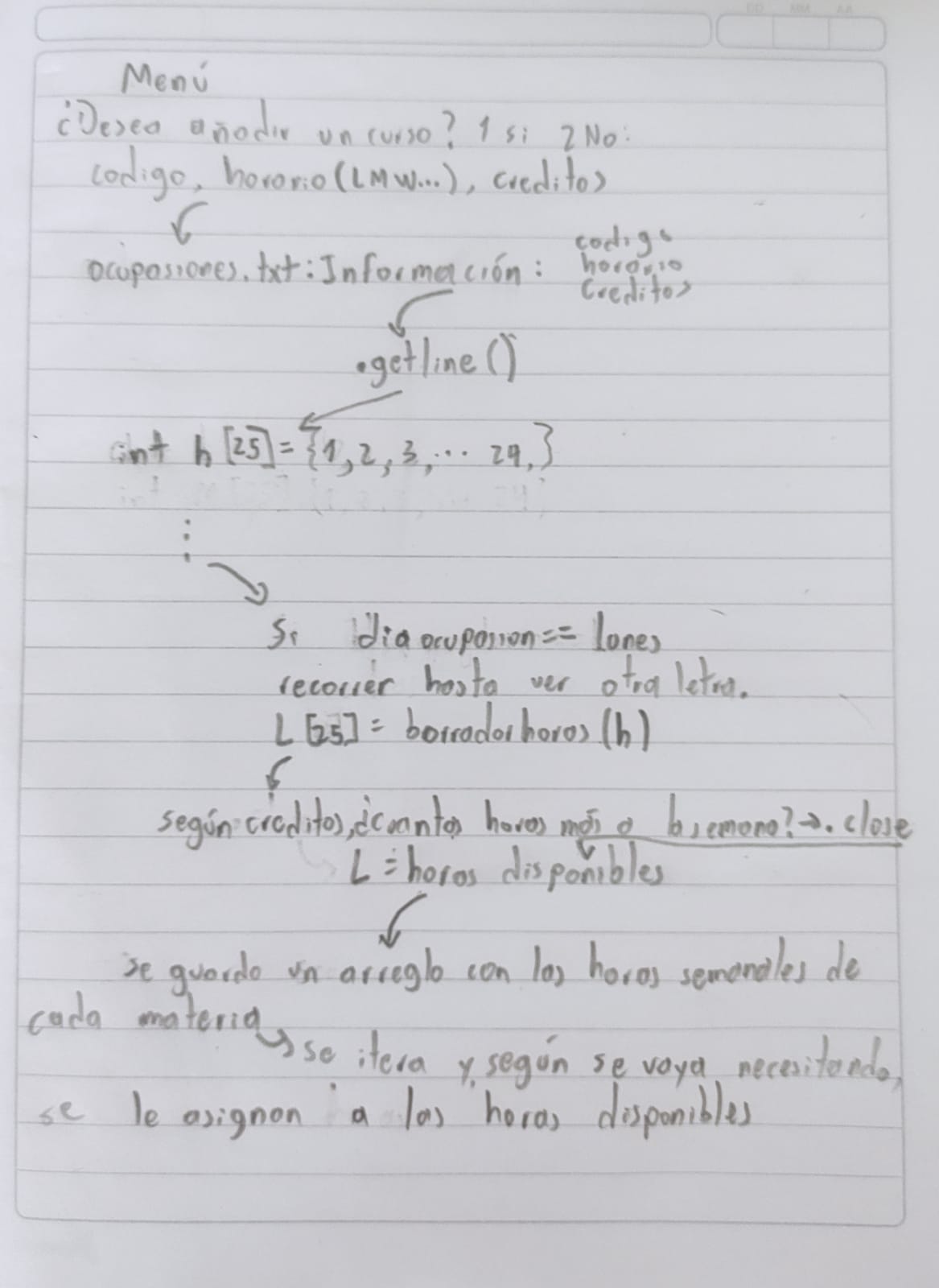
**ANALISIS GENERAL:**

* El parcial se basa en 3 cosas fundamentales: gestión de cursos, gestión de horarios y entrega de horario recomendado
* La idea es que el código trabaje de manera base con un archivo. En este archivo planeo contener toda la información de los cursos. El horario recomendando se imprimirá en la consola.
* El archivo me gustaría dividirlo en 2 con la siguiente información: primero, información de la universidad. Aquí planeo tener el código, el horario con profesores que se tienen en la semana y total de créditos. La segunda parte con complementos. En este pienso poner una especie de códigos que se construyen de manera muy distinta a los códigos de la universidad (puede ser enumerar actividades muy generales que normalmente nos pueden quitar tiempo como dormir, deporte, cocinar, etc. del 1 al 10) pero que igual tengan su horario y que sirva para modelar de mejor manera el horario.
* Para construir el archivo, le pediré al usuario que vaya ingresando la información y el código se encargará de que se vaya acoplando de la manera que yo como programador necesito para el buen trato de la información en el código.
* Se hará un menú para preguntarle el curso y su información hasta que el usuario ingrese a la opción de no dar más cursos, Luego entrará a un menú de complementos donde se le dará unas opciones de código de diferentes actividades generales que también puede hacer a la semana como dormir y deporte por ejemplo.
* Para leer el archivo, busco que cada parte de información este en un renglón o cambio de línea diferente, ya que así se me facilitaría mucho al momento de gestionar la información, pues con un getline() accedo a todo el renglón. Si yo sé que en el primer renglón es el código de la materia, el segundo sus créditos y el tercero su horario con profesores en la semana yo ya se cómo tengo que trabajar cada renglón, además si sigo este patrón con las otras materias, yo sé que la secuencia se repite cada 3 renglones. Para el complemento me gustaría hacer una separación con un renglón vacío o solo con un cambio de línea ‘\n’. Cuando se llegue allí sabre que la próxima línea seguirá el código de la actividad y la que le sigue su horario en la semana y así sucesivamente, muy parecido a la parte de las materias.
* Lo anterior es para la gestión de cursos. Ahora para la gestión de horarios se podría tener una distinción de días de la semana con su inicial como normalmente se hace en la universidad (es decir, cuando hay una ‘L’ ya se sabe que es lunes, ‘M’ martes y así sucesivamente), con la distinción de que el miércoles se representaría con ‘W’ para evitar confusiones con el martes. Cada día sea como un arreglo de enteros que contenga los números del 1 al 24. Esto seria como una matriz donde las filas podrían representar los días de la semana y las columnas las horas del día. La matriz podría ser tipo booliana donde nos indique con un false si esa hora esta disponible y con un true si esta ocupada. Si esta disponible ya sabemos que podemos disponer de esta hora, se imprimirá dicha hora con la materia a la que se le asigno y luego se cambiara el estado de false a true para evitar que vuelvan hacer uso de este horario. Además, llegado el caso de que dos horarios que nos dio el usuario son la misma hora, con esta forma de llevar los horarios, sabre que habrá un error y se podrá notificar de que hay dos horas que tienen la misma materia, lo cual no puede llegar a pasar.
* Para la parte de entregar el horario, tendríamos que hacer un plan para distribuir las horas de estudio según las horas que nos sobran. Primero pienso sumar todas las horas de los diferentes cursos de los cuales el estudiante debe de hacer uso a la semana para cumplir con la intensidad horaria (trabajo a parte de las horas con el profesor. Luego de eso, según esas horas por día, distribuirlas por cada día de la semana. Segun la matriz general que tenemos, recorrerla y si esa posición es falsa(que está libre) Se le asigna esa hora a la materia y se cambia a true(ocupada. Si en algún día no se pueden completar las horas, se recargarán al día siguiente y así sucesivamente. Si se llega a todos los días de la semana y aún hay recargo de horas (horas que no se pudieron acumular) el programa lanzara una alerta donde diga que se recomienda cancelar algún curso o actividad complementaria porque el tiempo no le va a dar. Si se dan los recargos de manera exitosa y se pudo distribuir todo entonces el programa ira iterando por cada día de la semana imprimiendo cada hora de estudio y al curso que se enfocara esa hora. Es importante también tener en cuenta a la intensidad horaria de cada grupo de manera individual para saber cuánto se le debe de dedicar en un día a cada materia y así distribuirla de buena manera.
* **NOTA:** la parte de los complementos es una forma de que el código sea mas realista y quede mejor modelado, sin embargo, no será la prioridad. Se empezará a desarrollar cuando la parte base del trabajo este completa y funcional.

**FUNCIONES PLANEADAS:**

* Una función que me permita, a partir de los créditos, obtener la intensidad horaria semanal de cada materia.
* Una función para obtener las longitudes de las cadenas de caracteres.
* Una función para eliminar un elemento en específico de una cadena de caracteres
* Una función que a partir de un arreglo con horas disponibles lo modifique para que solo queden las horas que se van a utilizar de las materias
* Una función que me retorne la cantidad de reglones que contiene el archivo.
* Potencia: Función que recibe un entero a que es la base y un entero b que es el exponente. Retorna la potenciación. Sirve para otra función que pasa de char a enteros.
* Solonumeros: función que recibe una cadena de caracteres y retorna un false si hay algún carácter diferente a un numero y true si todos los caracteres son números.
* Longitud: Función que recibe un arreglo y retorna su longitud(longitud de elementos importantes, no elementos vacíos).
* Conversorcharint: Función que recibe una cadena de caracteres numérico y retorna su valor en tipo int.
* Semana: función que retorna un true si lo que se ingreso es la inicial con mayúscula de un día de la semana.
* comparador:compara dos cadenas de caracteres y retorna verdadero si son iguales y falso si no lo son.
* Verificadorcodigo: es una función booliana que recibe una cadena de caracteres con un código y acá se buscara en una base de datos. Retorna true si esta y false si no.
* Acomodador: esta función no recibe ni retorna nada. Se encarga de preguntarle al usuario el código del curso y sus horarios, para acomodarlos de la manera en que lo necesito en un archivo.
* Matrizhoras: recibe un arreglo char que contiene la línea con los horarios y también recibe la matriz de la semana. No retorna nada, solo modifica la matriz según los horarios que recibió.
* Getname: recibe dos arreglos tipo char, el primero donde esta el código y el segundo es para guardar el nombre del curso según el código y la base de datos.
* Asignación: retorna un booliano, si se pudo asignar todas las horas retorna true, sino retorna false. Recibe las horas de estudio con el proferos, arreglo char con el código de la materia, y un arreglo char donde vamos a guardar el nombre del curso.
* Asignacionusuario: Es lo mismo que el anterior pero adentro de la función esta modificada para que el usuario elija las horas.
* Horadocente:recibe una cadena de caracteres donde esta el horario y según eso cuenta las horas con el profesor semanales. Retorna esas horas tipo int.
* Horasobrantes:recibe dos enteros, uno con los créditos y otro con las horas semanales con profes. Retorna un entero con la cantidad de horas de estudio independiente.
* Getcredito: recibe dos cadenas de caracteres, con el código y otro donde se va almacenar la cantidad de créditos según su código. No retorna nada.

**ESQUEMA QUE MUESTRA DE MANERA MAS GRAFICA LO QUE PIENSO HACER:**



**EVOLUCION DEL ANALISIS:**

* Antes estaba pensando en hacer arreglos de enteros por cada semana, pero al plantearme el problema de que pasaría si el usuario ingresa días y horas iguales de horarios de materias, como notificar el problema y se me vino a la mente que la mejor solución es trabajar con una matriz booliana de espacios donde itere del 1 al 7(días de la semana) y me diga con true o false si está disponible esa hora o no.
* No había entendido bien lo del primer módulo. Por lo que se cambia un poco la manera en que se toma la información. Ya no se le pedirá al usuario que ingrese los créditos de la materia, ya que habrá otro archivo donde según el código se tiene el nombre de la materia y los créditos. Al usuario si se le pide el código, pero es para compararlo con la base de datos y también se le pide el horario de esa materia tal y como lo tenia pensado. De resto todo se hace igual.
* Se limito la cantidad de horas de 24 a 17. Esto para hacer un poco más real la asignación de horas y respetando horas de sueño.
* Cuando termine el trabajo base, intentando mejorar los resultados y que todo se viera mejor, me encontré con el problema de que las horas asignadas de estudio independiente y las horas con el docente ya estipuladas se representaban de la misma manera en el horario que se imprime al final. Es por esto por lo que quise cambiarlo, pero al tener la matriz como un booliana se me dificultaba, pues solo había 2 estados. Ocupado o libre. Sin embargo, pensé en cambiarla a tipo int, pues allí podre tener mas de dos estados. Es así como definí los espacios libres como 0, horas con profesor 1 y horas asignadas como 2.
* Use memoria dinámica para reservar memoria. Sin embargo, como nunca sabía que tamaño iba a tener le asigne gran memoria y la borro cuando la dejo de utilizar para optimizar lo mas posible la memoria. Me hubiera gustado administrarla mejor pero no encontré la forma solo usando chars.
* La parte que quería realizar con actividades adicionales como dormir y hacer deporte no lo vi viable para el tiempo que tenía, así que para hacerlo mas real y con esta limitante opte por limitar el tiempo del horario de las 6 am a 10 pm.
* Como agregado le puse la opción de decir que no le gusta el horario y dejo que el usuario agregue las horas de cada materia como lo desee.