**PROYECTO TERCER PARCIAL**

**Alumno: Máximo Diego Moreno Guerra**

**Grupo: 230B**

**Semestre: 2do Semestre**

**Modulo II**

**MODULO II. LENGUAJE C**

**Índice**

* **Introducción PAG. 3**
* **Uso básico del programa (abrir, cerrar y comandos básicos) PAG. 4**
* **Ejercicio 6 (SUMA\_PAR) PAG. 7**
* **Ejercicio 6 (Diagrama de flujo) PAG. 9**
* **Ejercicio 7 (PASSWORD) PAG. 10**
* **Ejercicio 7 (Diagrama de flujo) PAG. 13**
* **Ejercicio 18 (APROBE?) PAG. 15**
* **Ejercicio 18 (Diagrama de flujo) PAG. 17**
* **Ejercicio 21 (AGENDA) PAG. 18**
* **Ejercicio 21 (Diagrama de flujo) PAG. 22**
* **Notas PAG. 27**

**Introducción.**

El presente documento ofrece al lector (querido profesor) posee como propósito dar a entender las especificaciones y limitaciones del proyecto de “*Programación: Modulo II”* de 3er. Semestre, asi como explicar de forma detalla el cómo usar cada subprograma con imágenes reales de este mismo acompañado con diagramas de flujo para dar a entender mejor el funcionamiento de cada algoritmo realizado.

En total serán cuatro de cinco subprogramas los cuales funcionaran dentro de un programa principal, donde cada subprograma se dará una explicación de su algoritmo aplicado.

Es importante detallar la aplicación de técnicas más complejas para el mismo proyecto donde se busca expandir al máximo las capacidades de cada subprograma evitando problemas que puedan disminuir las mismas capacidades, y explorando técnicas interesantes para la creación de cada algoritmo explotando las mejores capacidades del lenguaje de programación ***C***.

Hay que aclarar que el quinto subprograma que se encargo fue trabajo libre, por lo tanto, no se incluirá en el presente documento.

**Uso básico del programa (abrir y cerrar)**

Interfaz de usuario gráfica

Descripción generada automáticamente Abrir el programa es de lo más sencillo que podrás hacer. Solo hay que hacer doble clic sobre el ejecutable “*ProyectoFinal.exe”* donde a continuación podrás probar todos los subprogramas realizados en este proyecto de fin de parcial.

Ilustración 1: Ejecutable .exe

Texto

Descripción generada automáticamenteAbriendo el programa, te encontraras con el siguiente menú:

Ilustración 2: Menú principal

Captura de pantalla de computadora

Descripción generada automáticamente con confianza mediaCada palabra que se encuentra entre comillas es el comando para poder acceder al subprograma. Aparte se deben escribir cada subprograma en MAYUSCULAS sin comillas en el apartado de “*Dame una de las opciones: “*. Sería algo así:

Ilustración 3: Escritura de comandos

Así podrás probar cada uno de los subprogramas disponibles. En caso de que el programa principal no logre reconocer lo que escribiste, podrás volver a intentarlo escribiendo en el apartado “Quieres probar algun otro subprograma?” dos de las siguientes opciones:

* “s” -> Sí
* Texto

  Descripción generada automáticamente“n” -> No

Ilustración 4: Comando no reconocido

Texto

Descripción generada automáticamenteSi se escribe “s” se podra volver a intentar escribir algun comando de un subprograma en específico. De lo contrario, si se escribe “n”, el programa se terminará, y para poder cerrarlo por completo, solo tendras que presionar alguna otra tecla.

Ilustración 5: Opción "s'

Texto

Descripción generada automáticamente

Ilustración 6: Opción "n"

Texto

Descripción generada automáticamenteSi se escribe cualquier otra cosa que no sea “s” o “n”, directamente se cerrará el programa terminando con su ejecución.

Ilustración 7: Opción no reconocida

Texto

Descripción generada automáticamentePara terminar este apartado, podrás escribir el comando “*AYUDA”* para poder volver a imprimir el menú de inicio cada vez que lo necesites para poder apoyarte con la descripción de cada subprograma con su respectivo comando.

Ilustración 8: Comando "AYUDA"

**Ejercicio 6 (SUMA\_PAR)**

Sumar pares entre 1 y N

Para poder probar el subprograma 6, se debe escribir el comando “*SUMA\_PAR”* para así poder probar las características de este mismo.

**Imagen que contiene naranja, calle, oscuro, firmar

Descripción generada automáticamente**

Ilustración 9: SUMA\_PAR

Aquí podrás escribir cualquier número desde el 0 hasta el 92500, donde a partir de este número, podrían surgir problemas con la sumatoria que se realizara acabo.

**Texto

Descripción generada automáticamente**

Ilustración 10: Limite 92500

**Texto

Descripción generada automáticamente**Basicamente, el programa sumara de par en par hasta llegar al número que tu le introduzcas, mostrando el resultado a continuación.

Ilustración 11: Sumatoria

En caso de introducir algun número negativo, saltará un mensaje curioso y la sumatoria no se realizará:

Texto

Descripción generada automáticamente

Ilustración 12: “Espera, son legales los números negativos aquí?"

**Diagrama

Descripción generada automáticamenteEjercicio 6 (Diagrama de flujo)**

\*Versión simplificada del algoritmo utilizado en C

**Ejercicio 7 (PASSWORD)**

Validar una contraseña.

Texto

Descripción generada automáticamentePara poder probar el subprograma actual, se debe escribir el comando “*PASSWORD”* con el cual, se imprimirá en la ventana del ejecutable lo siguiente:

Ilustración 13: PASSWORD

Para poder jugar, se deberá introducir cuatro numeros del cero al nueve (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) por cada intento, para poder adivinar la contraseña generada aleatoriamente por el mismo programa. Quedaría de la siguiente forma:

Texto

Descripción generada automáticamenteCada digito debe ser introducido individualmente respetando el rango del cero al nueve.

Ilustración 14: Introducir digito individualmente

Terminando de introducir los cuatro digitos, se imprimirá a continuación si cada digito es cierto o falso.

Si no se respeta el rango de los digitos que se pueden introducir, se podrían escribir una de dos alertas:

* Si el digito es menor a cero: “*Negativos nooo”*
* Texto

  Descripción generada automáticamenteSi el digito es mayor a nueve: “*Recuerda darme numeros del 0 al nueve!”*

Ilustración 15: Alertas

Captura de pantalla de computadora

Descripción generada automáticamenteEs importante detallar que se deben evitar introducir letras como si fueran numeros ya que, es caso de hacerlo, el programa colapsara y se terminara de golpe como se muestra en la siguiente ilustración:

Ilustración 16: Colapso del ejecutable

Para solucionarlo, deberá cerrar el ejecutable y volver a abrirlo.

Finalmente, en caso de haber ganado este, minijuego, se podria decir, se le notificara y el juego terminara.

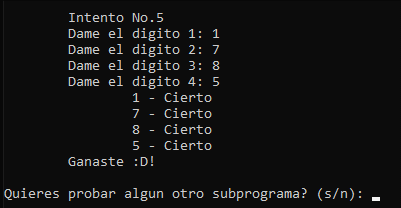
Independientemente de que intento haya ganado este minijuego, el subprograma terminara y podra probar algun otro subprograma. Recuerde escribir “*AYUDA”* en caso de querer volver a imprimir el menú de inicio.

Ilustración 17: Has ganado el minijuego!

Texto

Descripción generada automáticamenteEn caso de perder el minijuego, se imprimirá la contraseña digito por digito. No se espante si tarda en imprimirse, así esta programado la impresión de la contraseña para dar un poco más de, ¿suspenso?

Ilustración 18: Perdiste

Buena suerte intentando ganar este minijuego, la cual, realmente necesitaras. Cada que se ejecuta el subprograma se genera una nueva contraseña, por lo que tendras mucho tiempo para intentarlo.

**Ejercicio 7 (Diagrama de flujo)**

**Diagrama

Descripción generada automáticamente**

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

\*Versión simplificada del algoritmo utilizado en C

**Ejercicio 18 (APROBE?)**

Calcular la calificación del semestre a partir de tres notas, luego determinar si aprobó o no e informarlo

Texto

Descripción generada automáticamentePara probar este subprograma hay que escribir el comando “*APROBE?*” para asi utilizarlo dentro del ejecutable.

Ilustración 19: APROBE?

Texto

Descripción generada automáticamenteA continuación, te pedirá solo tres calificaciones debido a que un semestre se divide en 3 parciales (normalmente) y en base a estas mismas, se calculará el promedio debido para la calificación semestral.

Ilustración 20: Calificación Semestral

Se te indicará despues de calcular el promedio, si aprobaste o si reprobaste el semestre, y bajo de este mismo se mostrará tu promedio junto a uno de los siguientes mensajes:

* Promedio > 8: “*Que buen estudiante!”*
* Promedio > 6, Promedio < 8: “*Ya es algo!*”
* Promedio < 6: “*Bueno, a lo mejor, le intentaste! O tal vez no...”*

Texto

Descripción generada automáticamenteEn caso de introducir alguna calificación negativa, no se te calculara el promedio siendo este igual a cero, y se te mostrara una alerta (curiosa):

Ilustración 21: Calificación menor a cero

Texto

Descripción generada automáticamente  
Tambien mostrara una alerta distinta si introduces una calificación mayor a diez, tomando en cuenta el sistema educativo utilizado aquí en México:

Ilustración 22: Calificación mayor a diez

**Ejercicio 18 (Diagrama de flujo)**

Diagrama

Descripción generada automáticamente

\*Versión simplificada del algoritmo utilizado en C

**Ejercicio 21 (AGENDA)**

Crea un programa que simule una pequeña agenda

Texto

Descripción generada automáticamentePara poder utilizar el subprograma de la agenda, se debe escribir el comando “*AGENDA”* con el cual, se te mostrara el siguiente menú:

Ilustración 23: "AGENDA"

En el apartado “*Que haremos?”* se deberá escribir alguna de las cuatro letras en mayúscula haciendo cada uno actividad diferente para poder usar la agenda.

Iniciando con la letra “*N*”, esta letra nos permitirá añadir una nueva actividad para nuestra agenda, donde se deberán introducir los datos necesarios siendo:

* Nombre clave (máximo 12 caracteres, sin espacios)
* Hora (en formato 24hrs)
* Minutos de la hora anteriormente proporcionada
* Dia del mes
* Mes del año
* Año

Texto

Descripción generada automáticamente

Ilustración 24: Comando "N"

Algunos datos tienen limitaciones para evitar errores lógicos, siendo los siguientes:

* Hora -> Debe ser mayor a 0 y menor a 25
* Minutos -> Debe ser mayor a 0 y menor a 61
* Dia -> Debe ser mayor a 0 y menor a 31
* Mes -> Debe ser mayor a 0 y menor a 13

Texto

Descripción generada automáticamenteEn caso de que un dato no sea válido, se le volverá a pedir hasta que este mismo sea correctamente válido:

Ilustración 25: Validación de datos

Respecto a la validación del año, este es especial ya que no hay limitante como tal para poner tal numero, pero, si se mostrara dos *“easter eggs”* (secreto del programa) los cuales pueden ser:

Captura de pantalla de computadora

Descripción generada automáticamenteSi el año es menor al año actual (2024):

Ilustración 26: Chabelo

Texto

Descripción generada automáticamenteSi el año supero los cien años (2124):

Ilustración 27: Salud y Paciencia

Texto

Descripción generada automáticamenteAhora, pasando a la letra “*V*”, lo unico que hace es imprimir las actividades que usted lleva guardadas hasta el momento:

Ilustración 28: "Actividades guardadas"

Texto

Descripción generada automáticamenteLa letra “*B*” sirve para poder buscar alguna actividad en base a su nombre clave. Este comando es muy útil cuando tienes varias actividades guardadas. Te imprimirá el nombre clave, fecha y hora de realización:

Ilustración 29: "Búsqueda a partir del nombre clave"

Texto

Descripción generada automáticamenteEn caso de que el comando no encuentre ninguna actividad con el mismo nombre clave, simplemente saltara alerta indicando de la ausencia de la misma actividad:

Ilustración 30: No se encontró actividad con mismo nombre clave

Ya para ya casi acabar, la letra “*E*” cierra el subprograma de la agenda, terminando asi con las actividades. Nota: las actividades no se guardan en ningun lado, por lo que se recomienda hacer es guardarlas en algun otro lugar o archivo fuera el programa.

Ilustración 31: Salir de la Agenda

Texto

Descripción generada automáticamenteSi no se reconoce alguna letra de las ya vistas, puedes intentar escribiendo “*H*” para obtener ayuda de la Agenda.

Ilustración 32: Ayuda "H"

Tabla

Descripción generada automáticamente**Ejercicio 21 (Diagrama de flujo)**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Aplicación

Descripción generada automáticamente

Diagrama, Esquemático

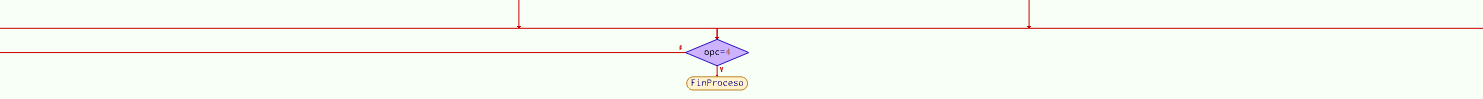
Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Tabla

Descripción generada automáticamente con confianza media



\*Versión simplificada del algoritmo utilizado en C

**Notas**

Interfaz de usuario gráfica

Descripción generada automáticamenteHay que aclarar que se mencionaron todos los errores posibles en cada ejercicio, y se les busco una solución en el codigo de **C** para evitar problemas, sin embargo, se pueden encontrar más errores los cuales pueden ser no mencionados aquí. En caso de que no se pueda cerrar el programa, abra el administrador de tareas, seleccione el programa “ProyectoFinal.exe”, de clic derecho y seleccione la opción “Finalizar tarea”.

Ilustración 33: PASO 2

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 34: PASO 1

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 35: PASO 3