

**Instituto Nacional de Aprendizaje**

**Curso**

**Programación de Aplicaciones Informáticas.**

**Módulo**

**Lógica computacional**

**Actividad**

**Práctica 11**

**Estudiante:**

**Mariel Daniela Rojas Sanchez**

**Profesor:**

**Oscar Pacheco Vásquez**

**2022**

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez.

|  |
| --- |
| **Planteamiento del Problema** |
| Crear un algoritmo donde el usuario pueda jugar al Cuadrado Mágico. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. Ingreso de 9 posiciones en la matriz. 2. 9 números ingresados en las posiciones. | 1. Mensaje indicando la matriz con posiciones. 2. Mensaje con la matriz de números que el usuario va ingresando. 3. Mensaje indicando si es o no un cuadrado mágico. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| * El cuadrado mágico se rellena únicamente con números del 1 al 9. * Se deben utilizar todos los números y estos no se pueden repetir. * Solo se considera cuadrado mágico cuando todos los elementos de cada fila suman lo mismo que la constante mágica, para este caso 15, así como los elementos de cada columna suman 15, y así mismo los elementos de ambas diagonales suman 15. * El cuadrado mágico es de orden 3. | |
|  | |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio. 2. Se definen las variables necesarias. 3. La constante mágica va a ser igual a 15. 4. Se les asignan números a las posiciones de la matriz, para que el usuario digite las posiciones donde desea agregar números. 5. Se imprimen la bienvenida y las instrucciones del juego. 6. En un subproceso se imprime la matriz con las posiciones. 7. Se hace un subproceso, con un repetir principal, dentro de ese repetir, se le solicita al usuario que digite la posición donde desea colocar el numero y se lee la posición. Se crea dos repetir dentro del repetir principal, con contadores, uno para las filas, y otro para que recorra las columnas, donde se verifica si la matriz2, es igual a la posición, para que se guarde el valor del numero ingresado en la matriz. 8. Se debe reiniciar el contador de las columnas para que este no cuente demás. 9. Se imprime la matriz con los números que el usuario va ingresando. 10. Con otros dos repetir, uno para fila y otro para columnas, se hace la búsqueda de todos los espacios vacíos, cuando no quede ningún espacio vacío, se puede hacer la verificación de las sumas de las filas, columnas y diagonal. 11. Se debe verificar si existen números repetidos en la matriz, si hay números repetidos, entonces no es un cuadrado mágico, si no hay números 12. Si las filas, columnas y la diagonal son iguales a la constante mágica (15), entonces se imprime en la pantalla si es un cuadrado mágico, si no, se imprime que no es un cuadrado mágico. 13. Fin de Algoritmo principal. | 1. Inicio. 2. Se definen las variables necesarias. 3. La constante mágica va a ser igual a 15. 4. Se les asignan números a las posiciones de la matriz, para que el usuario digite las posiciones donde desea agregar números. 5. Se imprimen la bienvenida y las instrucciones del juego. 6. En un subproceso se imprime la matriz con las posiciones. 7. Se hace un subproceso, con un repetir principal, dentro de ese repetir, se le solicita al usuario que digite la posición donde desea colocar el número y se lee la posición. Se crea dos ciclos “para” dentro del repetir principal, uno para las filas, y otro para que recorra las columnas, donde se verifica si la matriz2, es igual a la posición, para que se guarde el valor del número ingresado en la matriz. 8. Se debe reiniciar el contador de las columnas para que este no cuente demás. 9. Se imprime la matriz con los números que el usuario va ingresando. 10. Con otro sdos ciclos para, uno para filas y otro para columnas, se hace la búsqueda de todos los espacios vacíos, cuando no quede ningún espacio vacío, se puede hacer la verificación de las sumas de las filas, columnas y diagonal. 11. Se debe verificar si existen números repetidos en la matriz, si hay números repetidos, entonces no es un cuadrado mágico. 12. Si las filas, columnas y la diagonal son iguales a la constante mágica (15), entonces se imprime en la pantalla, si es un cuadrado mágico, si no, se imprime que no es un cuadrado mágico. 13. Fin de Algoritmo principal. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
|  | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa #1 | La alternativa 1 es mas eficaz, ya que al utilizar ciclos repetir, estos van a parar cuando ya se encuentre lo requerido, por lo que va a ser mucho optimo y rápido si se utilizaran ciclos para, donde estos siguen en ciclados hasta que terminen aun así si ya encontraron lo requerido. |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez.

|  |
| --- |
| **Planteamiento del Problema** |
| Crear un algoritmo que logre alcanzar la Constante de Kaprekar. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| Número ingresado. | 1. Mensaje indicando si se logró alcanzar la constante de Kaprekar, y cuantas restas fueron necesarias para llegar hasta ahí |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| Los dígitos del número ingresado no pueden ser todos iguales. | |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio. 2. Se definen las variables necesarias. 3. Se dimensión el vector de 4 espacios. 4. Se le solicita al usuario que ingrese un numero de cuatro dígitos. 5. Se lee el numero ingresado. 6. Se valida que en el numero no se repitan sus cuatro dígitos, si se repiten se imprime en pantalla que " El número ingresado repite sus dígitos.". 7. Se crea un subproceso Mayo, donde se va a acomodar el numero ingresado, con sus dígitos acomodados de mayor a menor para esto es necesario convertir el digito ingresado a texto, para poder ordenarlo por selección, y luego con un ciclo para que va recorriendo el vector, se convierte el digito mayor, con sus unidades de millar, luego el siguiente digito con sus centenas, el tercero con decenas y el digito más pequeño a unidades, cuando se convierten al final se concatenan ordenados de mayor a menor. 8. Se hace otro subproceso llamado Meno, que es igual al anterior pero ahora para acomodar los dígitos de menor a mayor. 9. En el algoritmo principal se crea un repetir, donde se va a llamar a imprimir el subproceso mayo y meno, para observar esos números ordenados de mayor a menor y viceversa, el número mayor se va a restar con el menor, y hay un contador que va a ir haciendo la sumatoria de cuantas veces se tienen que restar los números ordenados para llegar a la constante de kaprekar. 10. Si el numero que se va restando es igual a auxiliar, significa que el numero llego a la constante y que de ahí se va a seguir repitiendo. 11. Si el numero es igual a la auxiliar, se imprime en la pantalla cuantas veces se restó para llevar a la constante, y si no, se imprime que número no coincide con la constante de Kaprekar. 12. Fin de Algoritmo principal. | 1. Inicio. 2. Se definen las variables necesarias. 3. Se dimensión el vector de 4 espacios. 4. Se le solicita al usuario que ingrese un numero de cuatro dígitos. 5. Se lee el numero ingresado. 6. Se valida que en el número no se repitan sus cuatro dígitos, si se repiten se imprime en pantalla que " El número ingresado repite sus dígitos.". 7. Se va acomoda el numero ingresado, con sus dígitos acomodados de mayor a menor para esto es necesario convertir el digito ingresado a texto, para poder ordenarlo por selección, y luego con un ciclo para que vaya recorriendo el vector, se convierte el digito mayor, con sus unidades de millar, luego el siguiente digito con sus centenas, el tercero con decenas y el digito más pequeño a unidades, cuando se convierten al final se concatenan ordenados de mayor a menor. 8. Se repite el ciclo anterior que es igual al anterior pero ahora para acomodar los dígitos de menor a mayor. 9. Se crea un repetir, donde el número mayor se va a restar con el menor, y hay un contador que va a ir haciendo la sumatoria de cuantas veces se tienen que restar los números ordenados para llegar a la constante de kaprekar. 10. Si el número que se va restando es igual a auxiliar, significa que el numero llego a la constante y que de ahí se va a seguir repitiendo. 11. Si el número es igual a la auxiliar, se imprime en la pantalla cuantas veces se restó para llevar a la constante, y si no, se imprime que número no coincide con la constante de Kaprekar. 12. Fin de Algoritmo principal. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
|  | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa #1 | Se elige la alternativa 1, ya que, al estar conformado con subprocesos, el algoritmo se divide en partes más simples y se va resolviendo parte por parte, siendo el algoritmo más fácil de entender y trabajar |