**Informe de Trabajo Final de Promoción**

**Declaración de los tipos de datos utilizados en el programa**

subrOpciones = 'A' .. constOPCIONES; // Acotamos los valores de las variables para que en caso

subrCategorias = 1 .. constCATEGORIAS; de que ingresen valores no correspondientes no se tomen

en cuenta.

conjCompletadas = set of subrCategorias; // Declaramos un conjunto para guardar los colores de la

conjRespuestas = set of char; cara del diamante.

cadenaPreg = string[238]; // Utilizamos tres diferentes datos de tipo string con sus

cadenaOpci = string[210]; respectivas asignaciones de límite de caracteres para

cadenaColores = string[6] ; optimizar el uso de memoria.

arrOpciones = array [subrOpciones] of cadenaOpci; // Usamos un arreglo de strings para guardar las opciones de las preguntas

tipoPreguntas = record // Creamos un registro para las preguntas con sus respectivas respuestas y/o opciones con sus soluciones

numCategoria: subrCategorias;

numPregunta: integer;

pregunta: cadenaPreg;

opciones: arrOpciones;

respuesta: subrOpciones;

explicacion: cadenaOpci;

end;

lista = ^nodo; // Creamos una lista para guardar el registro de las

nodo = record preguntas en donde cada lista creada corresponde

datos: tipoPreguntas; a las categorías de la cara del diamante

sig: lista;

end;

tipoLista = record // Creamos dos punteros para agregar al final cada

pri, ult: lista; pregunta sea mas optimo en tiempo de ejecución

end;

vdlCategorias = array [subrCategorias] of tipoLista; //Creamos un arreglo de una dimensión para guardar las listas de preguntas de cada categoría de la cara del diamante

**Fundamento:**

Decidimos crear un arreglo de punteros (dato vdlCategorias) con el objetivo de asegurar la posibilidad a futuro de poder cambiar las preguntas disponibles en el juego sin necesidad de modificar el programa.

**Preguntas de cada categoría con su respectivas opciones, respuestas y justificaciones**

**█ Categoría Roja: “Conceptos introductorios: Datos, Variables”**

**Pregunta 1:** ¿Cual de estos es un tipo de dato no ordinal?.

1. Entero
2. Real
3. Carácter

**Respuesta:** El real es un tipo de dato no ordinal ya que tiene infinitos números entre sí, por lo tanto, no tiene un antecesor y un sucesor

**Pregunta 2: ¿**Cual operación devuelve el cociente entero de una división?.

1. MOD
2. DIV
3. /

**Respuesta:** La operación DIV es la operación encargada de devolver el cociente de la división entre 2 números.

**Pregunta 3:** Las constantes se declaran de la siguiente forma:

1. Type NombreConstante = ‘Valor’;
2. VAR NombreConstante: ‘Valor’;
3. Const NOMBRECONSTANTE = ‘Valor’;

**Respuesta:** En Pascal, las constantes se declaran bajo la palabra reservada const.

**Pregunta 4: ¿**Cuál de las siguientes opciones es la estructura de control de decisión?.

1. If (condición) then else
2. While (condición) do
3. For i:=1 to valor do

**Respuesta:** El if, por definición, es la estructura de control de decisión ya que pregunta una vez si una o más condiciones son verdaderas o no, el while es una estructura de control pre condicional y el for es una estructura de control de repetición

**Pregunta 5:** ¿Cuál de los siguientes tipos de variables no se puede usar como parámetro de la estructura de repetición:

1. Numérico
2. Real
3. Char

**Respuesta:** El real es el tipo de dato que no se puede utilizar en una variable de control ya que es un tipo de dato no ordinal.

**█ Categoría Verde: “Modularización, comunicación entre módulos, pasaje de parámetros”.**

**Pregunta 1:** ¿Qué tipo de datos puede devolver una función?

1. Datos Simples
2. Datos Compuestos
3. Estructuras de datos

**Respuesta:** Las funciones solo pueden devolver un tipo de dato simple y no puede ser de otro tipo, esto no es aconsejable, ya que las funciones se pueden utilizar dentro de las declaraciones de otras estructuras de control.

**Pregunta 2:** Para poder modificar los valores de una variable por un procedimiento se recomienda pasarla como:

1. Parámetro por valor
2. Como variable global
3. Parámetro por referencia

**Respuesta:** Siempre es recomendable pasar las variables como parámetros por referencia si se lo modifica. Los parámetros pasados por valor no modifican el parámetro real y las variables globales son más difíciles de seguir.

**Pregunta 3:** ¿Qué es un procedimiento?

1. Un módulo que realiza tareas y puede devolver 0, 1 o más variables a través del método de pasaje por parámetros
2. Un módulo que realiza tareas y solo devuelve un valor
3. Es una estrategia que implica dividir un problema en partes funcionalmente independientes

**Respuesta:** Un procedimiento no necesariamente debe devolver un solo valor, y la estrategia de subdividir un programa en partes funcionalmente independientes se conoce como modularización.

**Pregunta 4:** ¿Como se invoca una función?

1. Por medio de una estructura de control (asignación, comparación, etc.).
2. Nombre (variable1, variable2, etc);
3. Readln (Nombre (variable1, variable2, etc));

**Respuesta:** Una función puede ser declarada dentro de cualquier estructura de control que requiera de un dato simple.

**█ Color Azul: Organización de computadoras.**

**Pregunta 1:** Complete con la compuerta lógica correspondiente: 11111....01100 = 01100

1. OR
2. AND
3. XOR

**Respuesta:** Es AND porque cuando dos entradas es 1 la compuerta lógica te devuelve 1.

**Pregunta 2:** Cuáles de estas operaciones es la correcta

1. X XOR 1 = X
2. X AND 0 = X
3. X OR 0 = X

**Respuesta:** La respuesta es XOR ya que la máscara 1 te devuelve la misma entrada y con AND debe ser con 0 para que te devuelva la misma entrada y con OR basta que una de sus entradas sea 1 para que la función valga 1

**Pregunta 3: ¿**Cuántas combinaciones de números se pueden realizar con 8 bits en BSS?

1. 0..127
2. 0..255
3. -63...63

**Respuesta:** Para calcular el rango en BSS es de la siguiente forma 2^n-1 donde n es el número de bits y la respuesta es 255 ya que 2^8-1= 255.

**Pregunta 4:** La unidad aritmético lógica es la encargada de:

1. Encontrar la celda de memoria en la que se encuentra cada dato requerido.
2. Recibir datos, realizar operaciones lógicas y matemáticas a partir de instrucciones definidas con los mismos y devolver los resultados de dichas operaciones.
3. Recibir resultados de operaciones lógicas y reintegrarlos en su respectiva celda de memoria.

**Respuesta:** Ya que, como lo indica la definición de aritmética: Parte de la matemática que estudia los números y las operaciones que se hacen con ellos.

**Pregunta 5:** ¿Cómo se realiza la normalización en un sistema de punto flotante con mantisa fraccionaria?

1. Mover el punto fraccionario de manera tal que el bit más significativo después del punto (sin considerar el bit de signo) sea 1 y para no modificar el valor se compensa con modificar el exponente
2. Desplazar en forma dinámica la coma decimal a una posición conveniente y usar el exponente de base 10 para definir la ubicación de la coma
3. Ninguna de las anteriores

**Respuesta:** La normalización en un sistema de punto flotante con mantisa fraccionaria implica mover el punto fraccionario para que el bit más significativo sea 1, ajustando el exponente para mantener el valor original.

**█ Color Amarillo: Estructuras de datos, Registros.**

**Pregunta 1: ¿**Cuál de estas operaciones se puede aplicar a la variable registro?.

1. Suma
2. Asignación
3. Comparación

**Respuesta:** Solo se puede realizar la operación asignación en las variables del tipo registro, aunque esto no hable de las operaciones que se le puede aplicar a sus campos.

**Pregunta 2:** De acuerdo al tipo de datos que lo compone, el registro es una estructura de datos

1. Heterogénea
2. Homogénea
3. Numérica

**Respuesta:** El tipo de datos registro es una estructura de datos heterogénea ya que sus campos pueden estar compuestos de otros diversos tipos de datos.

**Pregunta 3:** ¿Cómo le asignamos un valor a un campo de un registro?

1. IdentificadorRegistro,campo := Valor;
2. IdentificadorRegistro.campo1 := Valor;
3. IdentificadorRegistro[campo] := Valor;

**Respuesta:** Solo se puede realizar la operación asignación directa en un registro de la siguiente forma IdentificadorRegistro.campo1 := Valor;.

**Pregunta 4:** ¿Cuándo decimos que una estructura de datos es de acceso directo?

1. Cuando se puede acceder a un elemento deseado recorriendo toda la estructura y sumando los elementos adyacentes.
2. Cuando se puede acceder a un elemento deseado recorriendo toda la estructura.
3. Cuando se puede acceder a un elemento deseado sin necesidad de recorrer toda la estructura.

**Respuesta:** Por definición, las estructuras de datos de acceso directo se pueden acceder a cualquier elemento que la compone sin recorrer la estructura, de ahí su nombre.

**Pregunta 5: ¿**Que estructura de control se utiliza para permitir el acceso de todos los campos de un registro directamente?.

1. With .. do
2. Repeat .. Until
3. While .. do

**Respuesta:** La estructura de control with .. do sirve para acceder directamente a los campos de un registro sin tener que llamar al registro constantemente, solo a sus campos.

**█ Color Morado: Arreglo, Listas y Eficiencia.**

**Pregunta 1: ¿**Qué es un arreglo?

1. Es una colección de elementos homogéneos a los que se accede mediante un índice.
2. Es una colección de elementos homogéneos con una relación lineal que los vincula
3. Es una estructura de datos heterogénea donde cada dato que lo compone se lo denomina Campo.

**Respuesta:** El tipo de dato arreglo es una colección de elementos a los que se puede acceder a sus datos mediante un índice.

**Pregunta 2:** ¿En qué consiste el método de corrección de programas “Walkthrough o Recorrido”?

1. El método consiste en recorrer un programa con alguna otra persona y calcular la cantidad exacta de memoria que ocupa cada variable.
2. El método consiste en recorrer un programa con alguna otra persona.
3. El método consiste en recorrer el programa instrucción por instrucción y compilarlo mentalmente.

**Respuesta:** Ya que el método walkthrough consiste en ver las instrucciones del programa paso por paso con una persona que no comparta los mismos preconceptos y pensamientos, que a su vez esté dispuesta a descubrir errores.

**Pregunta 3: ¿**Cuáles de estas operaciones no son permitidas para un tipo de dato puntero?.

1. Readln (puntero);
2. If (puntero1 = puntero2) then
3. Dispose(puntero);

**Respuesta:** No se puede realizar un Readln sobre un puntero ya que apunta o señala a la variable almacenada en la dirección de memoria que contiene el puntero.

**Pregunta 4: ¿**Cuando se considera correcto un programa?.

1. Cuando el programa compila.
2. Cuando el programa cumple la función especificada.
3. Cuando el programa está bien modularizado.

**Respuesta:** Ya que un programa se considera correcto cuando cumple la función que se le propuso, que un programa sea correcto no depende de su calidad, eficiencia o modularización.

**Pregunta 5:** En caso de perder el primer puntero de una lista:

1. No es posible acceder a la lista, pero la lista sigue cargada en memoria dinámica.
2. La lista automáticamente se libera en la memoria dinámica.
3. La lista automáticamente vuelve al primer nodo y sigue cargada en memoria dinámica.

**Respuesta:** Cuando no recordamos guardar el primer puntero de una lista se pierde el acceso a la misma ya que la lista es una estructura de datos de acceso secuencial, y no podemos acceder al inicio de la secuencia.

**Bibliografía Utilizada:**

- “Programación en Turbo Pascal” (originalmente “Programming with Turbo Pascal”) de David W. Carroll, traducido por Laura Magda Figueroa Peña.

- “Programación en Pascal” serie Schaum de Byron S. Gottfried.

- Power Points de teoría de la cátedra.

**Herramientas Usadas (links):**

- [Lazarus IDE](https://www.lazarus-ide.org/).

- [Github](https://github.com/).

- [ExcaliDraw](https://excalidraw.com/) (gráficos). [Gráfico Completo](https://excalidraw.com/" \l "json=FYHMOiAMAacl0YPDIuN5t,EuaPAl6GXMzd8cGCwIkSHQ).