Algoritmos y Programacion I - Mendez **(Fecha: 19/09/19)**

Trabajo Practico N1 - Helm Royale

Correctora: Sofia Carbon Posse **Nombre:** Luciano Salerno, **Padron:** 104906

1 Correcciones puntuales

1.1 Preguntar y validar

Las funciones de preguntar y validar estan bien para este primer trabajo practico.

- Hubiese estado mejor que ambas funciones preguntar() y validar() sean una.

  Como por ejemplo:

```
void preguntar_signo(int* signo){
    printf("1-De que signo sos?\n");
    scanf("%i", signo);
    while (*signo< MENOR_SIGNO || *signo MAYOR_SIGNO){
        printf("\tERROR. Reingrese un signo valido del 1 al 12.\n");
        scanf("%i", signo);
    }
}
```

1.2 Tipo de Signo

Esta funcion tambin esta correcta.

- Para tener en cuenta para mas adelante, una manera mas eficiente de resolverlo es utilizando % :

```
const int FUEGO = 1 ;
const int TIERRA = 2 ;
const int AIRE = 3 ;
const int AGUA = 0;

char tipo_sign(int *signo){
    int tipo_signo;
    tipo_signo = signo % 4;
    return tipo_sign;
}
```

  Observar que:

```
const int ARIES = 1;
const int LEO = 5;
const int SAGITARIO = 9;
```

  El operador % en c, al hacer $10\%2 = 0$, te devuelve el resto de la operacin, 10 dividido 2 es 5, y tenes resto 0. Entonces fijate que al hacer $9\%4$ la division es 2 con resto 1, analogo a $5\%4$ la division es 1 y el resto es 1. En el caso de $1\%4$,la division no se puede efectuar entonces por default queda el mismo numero, siendo 1. Fijate que se cumple en los cuatro casos que tenes.

1.3 Equipo

Esta funcion esta mal implementada.

- El ultimo else que tenes implementado, rompe con la modularizacion. Ademas, excede la responsabilidad de la funcion equipo, ya que se ocupa de elegir uno y ademas de volver a preguntar todo si no pudo asignarlo. De esto se deberia ocupar el main.
  Capaz la mejor manera era utilizar una estructura iterativa de control en el main.

- Esto no esta mal ni influye en la correccion, pero te lo comento como dato para que lo tengas en cuenta:

  ```
  const char ACCION_MAYUSCULA ='A';
  const char ACCION_MINUSCULA ='a';
  ```

  En vez de declarar las 2 , podes tener una sola y en la funcion chequear por ejemplo con la funcion **TOUPPER()**:

  ```
  #include <ctype.h> /

  printf("Ingrese una letra: ");
  scanf("%c", &letra);
  letra = toupper(letra);
  ```

  Asi como esta toupper(), hay muchas mas funciones que a veces hacen que el codigo sea mas legible. Podes investigarlas si te interesan.

1.4 Intensidad de mascotas

Esta funcion esta mal implementada.

- Haces un abuso de constantes:

  ```
  const int MASCOTA_SIMPLE = 1;
  const int MASCOTA_DOBLE = 2;
  const int MASCOTA_TRIPLE = 3;
  const int MASCOTA_CUADRUPLE = 4;
  const int MIN_MASCOTAS = 0;
  ```

- Los cases del switch son los famosos "MAGIC NUMBERS" que hay que tratar de evitar siempre que uno programa. Una mejor solucion podria ser:

  ```
  int intensidad_mascotas(int cant_mascotas){
      if (cant_mascotas >= MAX_MASCOTAS){
                  return MAX_MASCOTAS;
          }
          return cant_mascotas;
  }
  ```

2 Correcciones en general

2.1 A mejorar:

- No abusar del uso de constantes.
- No utilizar MAGIC NUMBERS.
- Identificar de que se encarga cada funcion, en especial el main, buscando siempre la modularizacion.

2.2 Esta bien:

- Codigo legible y bien indentado.
- Pre y Post condiciones.
- Buen flujo de codigo.

El resto del trabajo esta muy bien. No te desanimes al mirar todas estas correciones, son mas que nada para que la proxima la tengas super clara en estas cosas.

- **NOTA : 8 (OCHO).**

  Cualquier duda o pregunta que tengas no dudes en consultarme!